An approach for segmenting 3D LiDAR data using Multi-Volume grid structures

Selma Goga, Sergiu Nedevschi
Image Processing and Pattern Recognition Research Group, Computer Science Department,
Technical University of Cluj-Napoca, Romania
selma.goga@student.utcluj.ro, sergiu.nedevschi@cs.utcluj.ro

***Abstract*— This paper proposes a novel approach for segmenting and space partitioning data of sparse 3D LiDAR point clouds for autonomous driving tasks in urban environments. Our main focus is building a compact data representation which provides enough information for an accurate segmentation algorithm. We propose the use of an extension of elevation maps for automotive driving perception tasks which is capable of dealing with both protruding and hanging objects found in urban scenes like bridges, hanging road barrier, traffic tunnels, tree branches over road surface, and so on. For this we use a Multi-Volume grid representation of the environment. We apply a fast primary classifier in order to label the surface volumes as being part of the ground segment or of an object segment. Segmentation is performed on the object labeled data which is previously connected in a spatial graph structure using a height overlapping criterion. A comparison between the proposed method and the popular connected-components based segmentation method applied on an Elevation Map is performed in the end.**

Keywords—segmentation; lidar; multi-volume grid; overlapping intervals; elevation map; hanging objects; protruding objects; connected components;

I. Introduction

As autonomous driving systems become more and more complex, the need of having a compact and efficient representation of sensor data is felt. One operation often follows another one (e.g. classification follows segmentation; tracking follows segmentation or classification; prediction follows tracking; and so on). At the basis of this processing pipeline lies the segmentation stage which is part of the perception module of the autonomous car.

LiDAR data segmentation is a widely discussed topic nowadays. Segmentation is concerned with splitting the input data into similar featured segments for easier further analysis. It has been addressed from the beginning as a graph clustering problem. Its accuracy is directly dependent on the chosen type of data partitioning. In our approach we try to segment data using a connected components approach on a previously built spatial graph.

Partitioning 3D LiDAR data as an Elevation map structure is a very popular and efficient approach [9][7][11][12]. The idea behind such a structure is to store in each cell an elevation value. This value can be for example the maximum or minimum height taken from the points falling inside a cell. The elevation map structures have the advantages of being built and processed fast. They also reduce the memory demands of the initial data as the input is simplified after it is projected onto the grid.

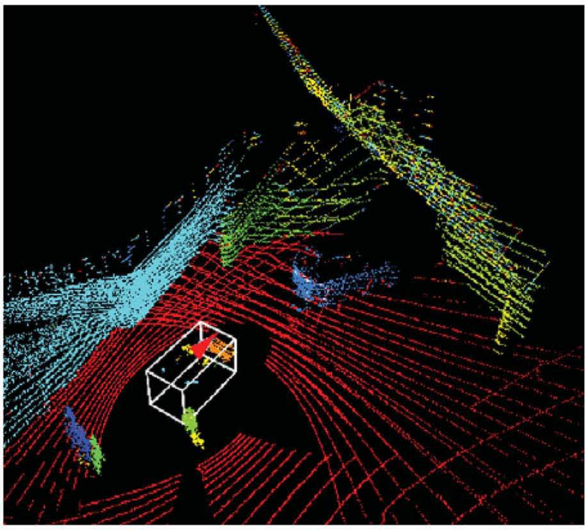

Fig. 1. Final segmentation result after creating the Multi-Volume grid and overlapping the volumes. The car with the four laser scanner sensors system is inside the wireframe box.

Unfortunately, Elevation maps have also some disadvantages which most of the time are the result of a poor segmentation. First, distinct objects that are too close will be eventually grouped together. This problem is directly dependent on the resolution of the grid. The second problem is that cells having multiple elevated objects (protruding or hanging objects) will not be noticed. Because of this, the set of points from distinct objects falling in the same cell will also be grouped together.

Hanging and protruding objects are a reality in nowadays complex urban environments. Bridges, hanging road barrier, traffic tunnels, tree branches over road surface or pedestrians, are just a part of the possible examples. This article tries to solve the problem with hanging and protruding objects using a multi volumetric grid partitioning structure of the space, letting the first described issue of Elevation maps as a future research topic.

A Multi-Volume grid[1] is an extension of Elevation Maps in a 3D perspective. The idea behind them is to store in each cell a list of elevation intervals. Each interval identifies the start and end heights of a *consistent vertical region*[2] found inside a given cell. This representation successfully solves the problem with hanging and protruding objects and assures a more accurate segmentation result as we will prove in the Experimental results section (IV).

This paper starts by outlining related literature approaches (II) and continues with describing our method (III) which is split in four main subchapters: Technology (III.A.), Multi-volume grid generation (III.B.), Ground segment extraction (III.C.) and finally, Segmentation (III.D.). The segmentation subchapter is again split in two sections: Building the 3D graph (III.D.1.) and Graph Labeling (III.D.2). We then show the experimental results of our work (IV) and sketch the conclusions (V) and further possible improvements (VI).

II. RELATED WORK

A solution to the problem of multiple elevated objects found inside a cell when building an Elevation Map has been discussed over a decade ago for mobile robots which used 2D range sensors for mapping the environment. An extension of Elevation Maps in order to solve this issue has been suggested in article [5]. Same authors proposed the creation of a Multi-Level Surface (a.k.a. MLS) map in article [3] in order to capture the tridimensional structure of a cell. The authors find consecutive vertical patch intervals inside a cell and compute and store the points height variance and mean values for each patch. Another studied solution found in literature is using a Multi-Volume Occupancy grid[1] first introduced for mapping the environment using an airborne LiDAR. The authors search for close points intervals in a cell based on the points height as in MLS[3]. They define an interval as the minimum and maximum height of a cluster of points with small height distance between consecutive elevated points as previously described in [3]. Each cell contains a list of such interval regions. These intervals can be seen as rectangular volumes with fixed base dimension (equivalent to the size of a cell of the grid) and variable height (based on the size of the consistent cluster of points). The same structure lies behind our proposed representation. The authors in [1] also compute an occupancy mass and density value which we decided to ignore for our current solution. Although all these solutions were proposed for navigation and environmental mapping on a mobile robot, we try to adapt them for our concern: handling the problem with protruding and hanging objects in Elevation maps for autonomous driving perception tasks. For this we decided using a modified version of the Multi-Volume grid space representation[1] for 3D LiDAR data in order to obtain a more accurate segmentation when multiple elevated objects are present inside a cell.

Another interesting idea is proposed in [2] which exploits the possibility of storing the height of points in histograms per each Digital Elevation Map cell. It then searches for vertical consistencies in histograms based on the number of gaps between two occupied adjacent buckets in order to determine the cells with a vertical profile. These vertical profiled cells are sure to have points coming from an object. Although the article was written for data coming from a stereo sensor, the ideas can be adapted and further used for improving the current implementation. This could be a possible future research topic.

Another possible solution, popular in stereo vision sensors, is considering only the points which are below a certain threshold [2]. This is not so efficient if the ground is not flat. Our proposed method is independent on the ground profile. An improvement of the previously presented technique to cope with such situations was proposed in [10]. It first finds the roads inclination, and cuts the points over a threshold based on the resulted shape afterwards. Nevertheless, all these methods prove to give good results in the end, but we are more concerned on building a compact space partitioning structure which can be sent for further analysis, without additional concerns.

3D data segmentation is a widely discussed topic nowadays because it is an important preprocessing phase in many intelligent vehicle tasks (e.g. object detection[9], SLAM[8], tracking[4][7][8]). The accuracy and speed of the segmentation influences directly the performance of the next analysis the system has to carry out.

Segmentation has been addressed from the beginning as a graph clustering problem. The most popular way of segmenting grid structures is done by applying a connected components based segmentation on the object labeled cells after a prior ground extraction took place [9][7][11][12]. This popular method has proven efficient in open spaces with low or rare vegetation and with no protruding or hanging objects. Our method builds a spatial graph with nodes being object labeled volumes and applies connected components in order to segment the data. A comparison is made in the end between the popular method of segmentation applied on an Elevation Map and our proposed method. Our method has also the advantage of working in systems with data coming from one or more fused LiDAR sensors as we prove in section IV.

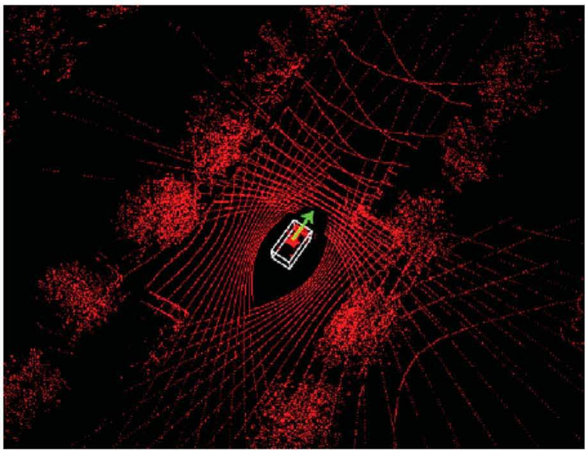

Fig. 2. Fused and aligned raw point clouds taken from the four separate 16 layered laser scanner sensors mounted on top of the car.

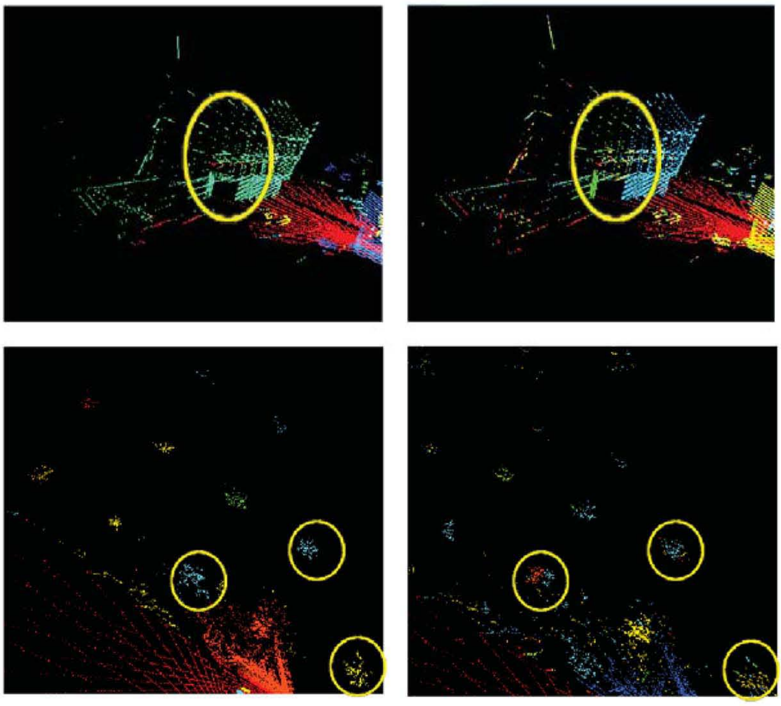

Fig. 3. The left column images use the variable version of *CellsDistTH* threshold while the right ones, a fixed threshold. The first row shows a building which is under-segmented in the second column. Same happens with the trees from the second row when a fixed threshold is applied. The improvement the threshold function brings is visible.

III. Proposed methodology

We define a *frame* as a cloud of points coming from a complete 360 spin of a Lidar sensor. For the system with four 16 layered LiDARS, a *frame* represents a fused cloud of points coming from the complete spin of all four laser scanners (see figure 2). The following described method is assumed to work on a single such frame.

The ground is not assumed to be flat for the proposed data partitioning structure although the used ground extraction method in this paper is limited. When a slope is present on the terrain surface it will most likely be seen as being part of an object although it is part of the road segment. A refinement stage could be added in order to solve this issue as proposed in [13].

Because this articles' main focus is on the segmentation of objects using a volumetric representation of the data, the refinement for the ground segment will not be discussed in this paper.

Note than the Z axis represents the height and is pointing upwards.

A. Technology

For scan acquisition we use four fused 16 layered laser scanners. Each sensor has a 360 degrees' horizontal field of view. All these sensors are mounted on top of the car. Each sensor has a different inclination angle against the ground.

An example of the fused registered point cloud is shown in figure 2.

All sensors have their data synchronized, aligned and motion corrected. These topics will not be discussed in the current paper.

We define the *origin* of the system in the center of the four sensors. If only one laser scanner is used, the *origin* is represented by the geometric center of that sensor. The distance from the origin to the ground VELO_H is known in both cases.

The implemented method is tested on the fused point clouds coming from all the four LiDARs. For testing the segmentation algorithm, we also used frames from the KITTI dataset taken with a HDL-64E Velodyne sensor. The accuracy of the proposed method was evaluated on the latter set of frames.

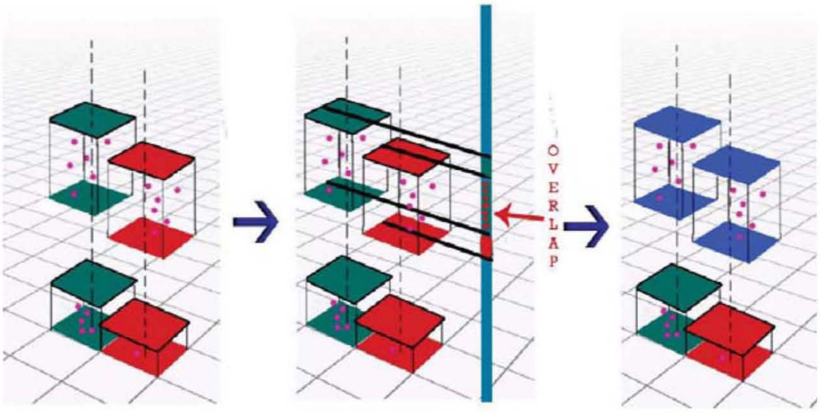

Fig. 4. The overlapping criterion is shown intuitively. The left image shows neighboring cells with their volumes. In the center image, a comparison of the two neighboring volumes height intervals is made in order to check if they overlap. Because in the above example the intervals do overlap, they are considered being part of the same segment and they receive the same label as seen in the right image.

B. Multi-Volume Grid Generation

We start by projecting the points onto a 2.5D top view grid. For our implementation the grid has a resolution of 16cm. For memory efficiency we store the array indices of the input point cloud for each cell. A nonempty cell will thus have an array containing indices of the points which fall inside it. We then order the array indices of each cell in ascending order based on the height of each corresponding point. This step is vital for building our final volumetric structure.

After all previous steps finished, we start adding the relevant 3D information per cell which will allow us to deal with both protruding and hanging objects. A *volume* is defined as having a fixed base (which is the dimension of the cell it is part of) and a variable height (which is defined as an interval with the bottom height and top height of the volume as endpoints). This is a simplification from the original proposed method from article [1] which also computed the occupancy mass and density value of points which fall inside a volume. Because for the current implementation we do not need these values, we ignore them. We describe the creation of the Multi-Volume grid structure through the following steps which are applicable for each cell:

- We start from the minimum elevation point found in a non-empty cell and store its height value as a bottom value of the first volumetric structure found inside the cell.
- We compute the height difference between it and the next point knowing that for each cell, all points inside them have been previously set in ascending order based on their elevation. If the difference exceeds GAP_TH, then the value of the height of the current point is stored as top value for the previously started interval. The generation of a volume is thus completed. The value of the height of the next point from the cell, if any, will be a bottom value for the next volumetric structure.

This method is also described in more detail in [3] where the authors focus on storing the variance and mean of points falling inside a volume rather than the geometric limits of it.

Because the point cloud is a sparse one, we allow the creation of volume shapes containing only one point. This occurs when we have a cell with only one point or when a point cannot be considered connected with any others neighboring points found inside the cell it falls in because of the gap distance.

C. Ground Segment Extraction

Ground segment extraction is a common pre-phase of segmentation algorithms because it eases the process of finding each object cluster afterwards.

The problem could be formulated as follows: given the initial point cloud input which is now represented as a set of volumetric shapes we want to assign a label to each of these volumes as either: ground or object.

From observations we can make the following assumption: the lowest height volume from a cell is either coming from an object or from the ground segment. If there is a cell which has more than one volume, all volumes except the first one are sure to be part of objects from the scene. Although the lase scanner sensor we used has a high accuracy rate, noisy data is present. Because the problem with noisy points found under the ground plane is well known and can infer with our previous statements, we have filtered the initial point cloud and have ignored the points which are 30 cm under the VELO_H constant.

After identifying the lowest height volumes of each cell, we filter them by applying a fixed threshold, MAX_GROUND_TH, on the top height value of each volume. All the volumes which are most likely from the ground, based on the result after applying the filter, are now extracted and the

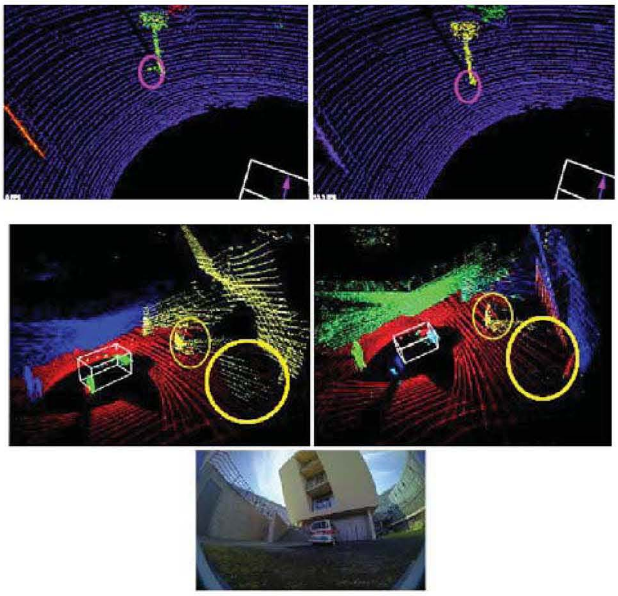

Fig. 5. The comparison between the popular segmentation algorithm applied on an Elevation Map (the left images from the first two rows) and our segmentation method applied on the Multi-Volume grid structure is shown (the right images from the first two rows). The last image shows the front camera view of the car from the system with four laser scanners for a deeper understanding of the scene.

remaining volumes are labeled as objects. This method is a fast and easy one and it gives great results for start as the topic of ground extraction is not the main concern of this paper. Notice that our proposed solution for ground segment identification does not work well in terrains with slopes. A refinement stage could be further applied as in [13] in order to solve this issue.

```
procedure create3DGraph()
 for each ci in Grid Cells
  if ci.label = common_object_label
   for n in 8-connected neighborhood
    getNextValidNeighbor(cj,n)
    if cj.label = object_label AND grow(ci,cj)
     for each nvi in ci.volumes
      for each nvj in cj.volumes
       if overlappingI (nvi, nvj) OR closeI (nvi,nvj)
        G.addEdge(nvi,nvj)
```

Fig. 6. The pseudocode for building the spatial graph. Grow() is a Boolean function which uses $CellsDistTH$ threshold in order to filter its input. CloseIntervs() is also a Boolean function and uses $CloseIntervalTH$ threshold in order to give the final result.

D. Segmentation

Having the set of previously labeled volumetric shapes, we want to cluster them and assign a unique label to each generated cluster.

We formulate this issue as a graph clustering problem.

1. Building the 3D Graph

We want to build a unidirectional graph *G* = (*Nv, Ev*), where:

- *Nvi* is a node represented by a volumetric shape, where i goes from 0 to the total number of volumetric shapes.
- *E* = (*Nvi,Nvj*) is the edge connecting two volumetric shapes *vi*, *vj*.

We use a connected components approach to generate the final graph *G*. For connections we have chosen an 8-connected neighborhood.

We say a cell Ci is a neighbor of the cell Cj if and only if:

- its neighboring cell Cj is the first nonempty cell in a chosen direction.
- neighbor cell Cj has been previously labeled as object.
- the distance between them does not exceed a certain threshold *CellsDistTH.*

Because of the problem with points density which decreases with their increased distance from the sensor we have made *CellsDistTH* a threshold function based on a

Segmentation method	Cloud partitioning structure	**Accuracy**
Connected Components on cells	Elevation Map	**94.9707 %**
Connected Components on volumetric shapes	Multi-Volume Grid	**97.6372 %**

Fig. 7. Comparison between the accuracy of the two tested segmentation methods. The first row refers to the popular segmentation method using a typical Elevation Map as an underlying structure. The second row shows the result obtained from our proposed method which is applied on the Multi Volume grid representation. Note that the accuracy is increased with 2.66 % when using the volumetric space partitioning structure.

variable *x*, where *x* is the Euclidean distance from the origin of the system to the current cell center Ci.

$$CellsDistTH(x) = 0.2 + \frac{1}{0.2+e^{-\frac{x}{7}+2.6}} \qquad (1)$$

The sigmoid function was generated from observations for our sensor system and it brought real improvement as figure 3 shows. We noticed that the same function can be applied successfully in the case of the data coming from one 64 layered Velodyne sensor.

When two neighboring cells are found, we try to connect (if possible) the volumes found inside them in order to create the spatial graph. For this we used a height overlapping criterion which states that two volumes are connected if their height intervals overlap. Figure 4 illustrates the intuitive approach behind this idea for a better understanding.

Because of the sparse data, there are nodes with only one point. These are not few so we had to take them into account. For this we added as refinement the possibility of neighboring node volumes to be linked together if the distance between their closest end points is smaller than a variable threshold $CloseIntervalTH$. This threshold is a function because we found that the density of points varies also on the vertical axis of the scene and not only on the distance between the points and the origin. We found that as the point is higher, the Euclidean distance between it and the closest above point tends to increase. We noticed that this distance increases with a constant rate. This is the reason why we have represented $CloseIntervalTH$ as a linear function as in equation (2).

$$CloseIntervalTH\ (\mathrm{x}) = 0.15 + \frac{x}{10} \qquad (2)$$

Note that two volumes found in the same cell can be considered connected only through their neighbors.

The pseudocode of the algorithm used for creating the 3D graph is illustrated in figure 6.

1. *Graph Labeling*

The final step is to label the object volumes based on the segment they belong to. For this we apply a connected components algorithm in order to identify and assign a unique label to each subgraph from the previously built graph *G*.

IV. Experimental Results

We have evaluated our algorithm on a series of frames captured in urban environments coming from two different types of laser scanner systems: one had four synchronized 16 layered laser scanners, and the other had only one HDL-64E Velodyne sensor. For testing we have used the following parameters for both systems: GAP_TH = 0.4m, MAX_GROUND_TH = 0.3m, VELO_H = 1.6m.

The program was written in C++ and ran on a computer with Intel Core i7, 3.50GHZ processor. The current application runs on a single thread and no major optimizations have been currently implemented. The average time performance for the segmentation algorithm without the generation of the grid and the extraction of the ground segment is 0.127 seconds for the Velodyne HDL-64E and 0.076 for the system with the four fused laser scanner sensors. We expect a multi-threading version of our algorithm to work faster. We have also implemented the classical segmentation method on an Elevation Map. This algorithm took 0.08 seconds per frames to process on the data coming from the 64 layered sensor and 0.052 when applied on the data from the four 16 layered sensors. The reason why the time performance is greater for the proposed algorithm is that the number of volumes is greater or equal to the number of cells. If the scene has hanging or protruding objects, the number of volumes will surely be greater thus the number of nodes in graph will be larger. More data will be processed in the end.

A visual comparison between our method and the Elevation Map method when protruding and hanging objects are present in scenes can be seen in figure 5. The first row displays the comparison between the popular literature method using an Elevation Map representation and our method and uses data coming from one HDL-64E sensor. Here, a part of a traffic sign projects directly onto the road segment. Problem is solved when the volumetric representation is used as it can be seen. The second row uses data coming from the four fused laser scanner sensors and makes the comparison described earlier. In this scene, the building surface is inclined over the road and the parked car. When using an Elevation Map representation, the road segment and the car under it is connected into one segment, the buildings' segment. The segmentation algorithm applied on a Multi-Volume grid proves its efficiency because it identifies each object correctly as it can be seen.

To compute the accuracy of the system for the proposed segmentation algorithm we have used a pre−labeled sequence of point clouds from the KITTI dataset. The clouds had labeled only cars, pedestrians and traffic signs, the vegetation, road segment and buildings being ignored in this measures. The 150 distinct clouds were coming from a HDL−64E Velodyne sensor. The final results can be seen in figure 7. As expected, the segmentation using the Multi−Volume grid gives better precision than the classical segmentation applied on a typical Elevation Map.

From observations we also got that the number of volumes with only one point are predominant above 3.5m height. The reason behind this is that the density of the LiDAR points is also affected by their elevation and not only by their range as stated in section III.D.1. A possible solution would be to considered the GAP_TH as a function with two parameters: first parameter being the distance from origin to a specific cell center and the second one, the height of a point above the ground surface. Another solution is to consider valid only the points which are below a given height threshold as in paper [2]. This solution would probably increase the time performance of the system because lower number of points would be processed.

V. Conclusions

The article proposed a new solution for data segmenting and space partitioning of sparse 3D LiDAR point clouds for complex urban environment scenes. The purpose was to increase the accuracy of the segmentation generated from a simple Elevation map by introducing the Multi− Volume grid structure which is an extension of elevation maps capable of capturing the tridimensional structure of a cell [1]. The volumetric grid map structure has successfully proven its efficiency compared to the classical Elevation Map structure as our proposed method increased the segmentation accuracy on our tested data with 2.66 %. This structures main strength is that it is capable of coping with both protruding and hanging objects found in urban scenarios. It is also independent on the roads' profile.

VI. Further Improvements

Several possible improvements have been outlined throughout the paper. We will remind them here:

1. the proposed solution doesn't address the problem of close neighboring objects. A possible solution for future research would be to allow the decomposition of a grid cell for a better precision[2].
2. The following described method is based on the LiDARS high accuracy in ranging. Although our implementation has been tested on frames with noisy ground 3D points and has given good results (figure 1 and figure 5), we consider applying in the future a noise filter on the raw point cloud.
3. Several fixed thresholds can become variable in order to increase accuracy.
4. A flexible extraction of the ground plane using this representation can be considered as a further research topic. A region growing approach as presented in [13] could be a good starting point.
5. The segmentation accuracy in case of using a polar grid has proven more efficient than on a rectangular grid[12]. Applying a volumetric structure on a polar grid and testing if the statement of the authors from [12] still stands is an interesting future research topic.

VII. Acknowledgements

This work has been supported by the UP-Drive project (Automated Urban Parking and Driving), Horizon 2020 EU funded, Grant Agreement Number 688652.

Special thanks to Robert Varga for the advice and support given when writing this article and to Ionut Pop for providing the collection of the ground-truth data and helping with the accuracy computation of the proposed method.

References

[1] Dryanovski I., Morris W., Xiao J., “Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles”, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice. New York: IEEE Press, 2010.

[2] S. Mandici, S. Nedevschi, “Real-Time Multi-Resolution Digital Elevation Map Creation using the Sensor Model of a Stereovision Sensor” , In 19th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2016.

[3] Triebel R., Pfaff P., Burgard W., “Multi-level surface maps for outdoor terrain mapping and loop closing”, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing. New York: IEEE Press, 2006.

[4] J. Rieken, R. Matthaei, M. Maurer, “Toward Perception-Driven Urban Environment Modeling for Automated Road Vehicles”, pp. 731-738, 2015.

[5] P. Pfaff and W. Burgard., “An efficient extension of elevation maps for outdoor terrain mapping”, In In Proc. of the Int. Conf. on Field and Service Robotics (FSR), pages 165–176, 2005.

[6] C. Rivadeneyra, M. Campbell, "Probabilistic multi-level maps from LIDAR data", In Int. J. Robot. Res., vol. 30, no. 12, pp. 1508-1526, Oct. 2011.

[7] A. Asvadi, P. Peixoto, and U. Nunes, “Detection and tracking of moving objects using 2.5D motion grids”, In 18th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2015.

[8] Jaebum Choi and Markus Maurer,”Local Volumetric Hybrid-Map-Based Simultaneous Localization and Mapping With Moving Object Tracking”, In 19th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2016.

[9] Asvadi A., Premebida C., Peixoto P., Nunes U., “3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes”, In Robot. Auton. Syst. 83, 299–311, 2016.

[10] F. Oniga, S. Nedevschi, M. M. Meinecke, T. B. To, "Road surface and obstacle detection based on elevation maps from dense stereo", In Proc. IEEE Intell. Transp. Syst. Conf., pp. 859-865, 2007.

[11] M. Himmelsbach, A. Müller, T. Lüttel, and H.-J. Wünsche, "LIDAR-based 3D object perception", In Proceedings of 1st International Workshop on Cognition for Technical Systems, München, Oct. 2008.

[12] D. Korchev, S. Cheng, Y. Owechko, and K. Kim, “On real time lidar data segmentation and classification”, In the 2013 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV’13), pp. 149–156, July 2013.

[13] S. Mandici and S. Nedevschi, “Aggregate road surface based environment representation using digital elevation maps,” In 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP) , pp. 149–156, Sept 2014.