Fast 3-D Urban Object Detection on Streaming Point Clouds

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. Efficient and fast object detection from continuously streamed 3-D point clouds has a major impact in many related research tasks, such as autonomous driving, self localization and mapping and understanding large scale environment. This paper presents a LIDAR-based framework, which provides fast detection of 3-D urban objects from point cloud sequences of a Velodyne HDL-64E terrestrial LIDAR scanner installed on a moving platform. The pipeline of our framework receives raw streams of 3-D data, and produces distinct groups of points which belong to different urban objects. In the proposed framework we present a simple, yet efficient hierarchical grid data structure and corresponding algorithms that significantly improve the processing speed of the object detection task. Furthermore, we show that this approach confidently handles streaming data, and provides a speedup of two orders of magnitude, with increased detection accuracy compared to a baseline connected component analysis algorithm.

Keywords: LIDAR, Urban Object Detection, 3-D Point Clouds, Dynamic Processing

1 Introduction

1.1 Problem statement

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. Vision systems with capability of handling continuously streamed sensor data have become important tools for robot perception [13]. Laser range sensors are particularly efficient

for these tasks since in contrast to conventional camera systems they are highly robust against illumination changes or weather conditions, and they may provide a larger field of view. Moreover, LIDAR mapping systems are able to rapidly acquire large-scale 3-D point cloud data for real-time vision, with jointly providing accurate 3-D geometrical information of the scene, and additional features about the reflection properties and compactness of the surfaces. The detection of urban objects is a fundamental problem in any perception motivated point cloud processing task [15]. Although it is a challenging problem itself, it can be helpful for several robot vision tasks, such as object recognition, localization or feature extraction. We focus here on the object detection problem relying on large-scale terrestrial urban point clouds, more specifically, we use point set data obtained by a Velodyne HDL-64 S2 laser acquisition system. The problem of detecting objects on streaming point clouds is challenging for various reasons. First, the raw sensor measurements are noisy. Second, the point density is uneven: [2] typically in terrestrial LIDAR point clouds the point densities dominate from the direction the measurement is taken, causing strongly corrupted geometric properties of the objects such as missing object parts or deformed shapes. The object detection process is further complicated when the data is continuously streamed from a laser sensor on a moving platform or a mobile robot. In this case we are forced to complete a complex task within a very limited time frame.

1.2   Related Works

A number of approaches are available in the literature for solving 3-D object detection and recognition problems in outdoor laser scans. The used data structure are essential part all of the existing techniques, and they can be coarsely divided into *two categories*.

   In the *first* category, traditional pre-computed tree-based data structures can be used, such as Kd-tree, Octree, range tree [3],[14]. These structures are efficient for performing range search, although there is a large processing overhead at initialization, and their performance rapidly degrades as updated data inserted after calling for reconstruction the tree structure [11]. Recent approaches apply different region growing techniques over tree-based structures to obtain coherent objects. The authors of [1] present an octree based occupancy grid representation to model the dynamic environment surrounding the vehicle and to detect moving objects based on inconsistencies between scans. However, the run-time and detection performance of the algorithm is not discussed here.

   The *second* category of the methods focus on grid-based data structures and efficient dynamic processing techniques for fast detection or recognition of objects from streaming 3-D data. In [7] the authors propose a fast segmentation of point clouds into objects, which is accomplished by a standard connected component algorithm in a 2-D occupancy grid, and object classification is done on the raw point cloud segments with 3-D shape descriptors and a SVM classifier. Different voxel grid structures are also widely used to complete various scene understanding tasks, including segmentation, detection and recognition [11]. The data is stored here in cubic voxels for efficient retrieval of the 3-D points.

Efficient range search from streaming data is an essential component of any object detection problem, and can be used for retrieval of all points which fall within a certain distance of a given point. For this task, a scrolling voxel grid data structure was proposed by [11]. The data is quantized here into small voxels of a prespecified resolution, then the indices of the voxels are shifted using a circular buffer according to the robot motion. To handle querying a large subvolume of space in sparse data, a sparse global grid was proposed by [8], when all streamed measurements were stored in a voxel-based global map. All of the approaches mentioned above provide convincing object detection results in large scale 3-D environment but they have some important limitations. Firstly, standard connected component solutions over tree-based data structures give very precise detection results, but they are not fast enough to serve real-time vision systems. Although, there exist efficient data structures for modifying minimum spanning trees which have sublinear complexity for each online update [4], this solution is impractical with streaming 3-D data [8]. Secondly, recent studies which suggest voxel, 2-D, scrolling, octree -grid based data structures for detection or recognition tasks do not propose optimal grid parameter settings (*e.g.* grid resolution or grid cell size) in order to minimize execution time, and maximize detection accuracy. Instead, they choose one certain grid resolution heuristically, and evaluate the performance of their detection method on this predefined grid resolution.

2 Proposed Approach

We propose a new data structure and a corresponding algorithm which is a basis of an efficient range search technique and a connected component analysis approach for fast object detection. In addition an optimal parameter setting strategy is proposed for enhancing the accuracy, which leads to the same or better detection performance than the tree-based approaches. More specifically, the following four main improvements have been implemented:

⋄ *Novel 2-D hierarchical grid structure for fast range search in 3-D:* a multi-level 2-D grid structure is presented with *two* different grid resolution levels (low and high). This structure is specifically designed for object detection *i.e.* connected component analysis tasks. We use these different grid levels to provide efficient and fast retrieval of 3-D point cloud features for the object detector module of our framework even in cases of strongly inhomogeneous point cloud density. We have experienced that standard 2-D grid structures [7] may give a decent result for region segmentation tasks *e.g.* ground detection, but they are not accurate enough near to the object boundaries, and they do not perform well in case of nearby urban objects. On one hand, using a large cell size multiple objects can occur within a given cell, resulting in several objects merged to the same extracted component. On the other hand a too dense grid structure (*i.e.* small cell size) may yield cells containing only a few points, which case does not enable us to calculate discriminative point cloud features for reliable classifica-
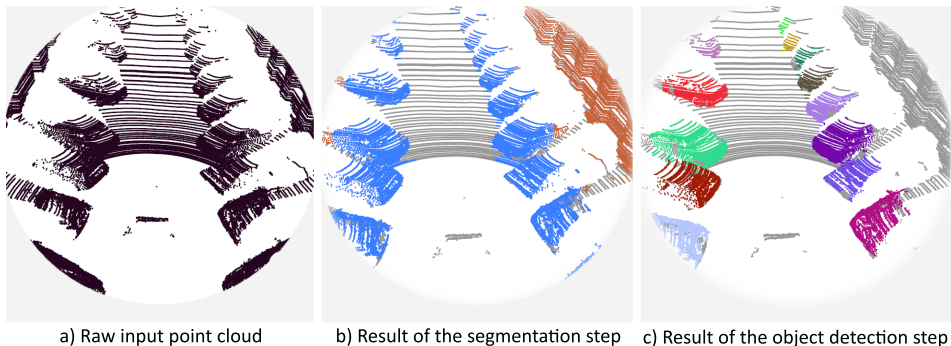
a) Raw input point cloud   b) Result of the segmentation step   c) Result of the object detection step

Fig. 1. Overview of the proposed object detection workflow. [Note: color codes in Fig. (b): brown = walls, grey=ground, blue=other street objects]

tion. In Section 3 we introduce the proposed grid structure in details.

⋄ *Connected Component Algorithm for streaming data:* a simple, yet efficient connected component analysis method is proposed in the hierarchical grid data structure, which provides reliable detection results in urban environment with real-time performance. In contrast to previous works [6],[14] this module of our framework queries local 3-D point cloud features from the hierarchical grid, and decides which 3-D points belong to the same urban object. The algorithm relies on different merging criteria to fulfill this task. See Section 3.2 for the details.

⋄ *Optimal grid resolution in urban environment:* In case of grid-based detection tasks, one of the biggest challenges is to find decent trade-off with respect to speed and accuracy. The major factor which can influence these properties is the grid resolution *i.e.* the size of a grid cell. It is crucial to select optimal grid resolution to keep the detection accuracy high, and the processing time low. In [7] the grid size has been selected manually without justification. Other approaches measure the entropies of the misclassification rate within the grid cell compared to different cell sizes. As a compromise to balance efficiency and accuracy they choose a certain grid resolution [8]. In contrast to above solutions, we propose a novel statistical metrics for approximation of the optimal grid resolution in terms of object detection.

⋄ *Publishing a new large dataset of hand-labeled 3-D point clouds:* We implemented a 3-D point cloud annotation tool for two reasons: First, we intend to provide a free annotated dataset to the research community. Second, using the Ground Truth (GT) we can evaluate the performance of our algorithm quantitatively, and we can compare it to earlier solutions.

The detailed description of the proposed object detection framework is structured as follows. In Section 3 we present a data structure that will allow us to

perform fast retrieval of 3-D point cloud features for segmentation and detection purposes. In Section 3.1 we describe our point cloud segmentation algorithm (see Fig. 1 b)). The point cloud is classified into large semantic regions such as *ground, walls, street objects* to prepare the data for object detection, which is presented in Section 3.2 (see Fig. 1 c)). We discuss the parameter sensitivity and the performance evaluation of the proposed grid model in Section 4 and 5.
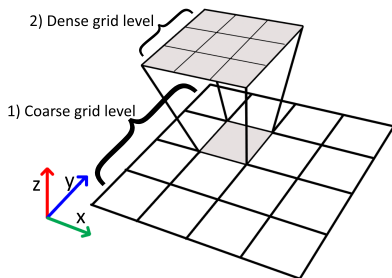
3   Data Structures

Fig. 2. Visualization of our *hierarchical grid model* data structure - *(bottom)* the coarse grid level: the 3-D space coarsely quantized into 2-D grid cells, *(top)* the dense grid level: each grid cell on the coarse level subdivided into smaller cells.

In this section, we introduce the grid based data structures used in the proposed system. First, we construct a *Simple Grid Model* [9] which will be used for initial point cloud segmentation, *i.e.* separating regions of roads, walls and short street objects. Second, we present a novel *Hierarchical Grid Model* which will be used for robust 3-D object detection from the strongly inhomogeneous density point clouds in challenging dense urban environments where several nearby object may be located close to each other.

$\diamond$ *Simple Grid Model:* We fit a regular 2-D grid S with W_S rectangle side length onto the $P_{z=0}$ plane (using the Velodyne sensor's vertical axis as the z direction and the sensor height as a reference coordinate), where $s \in S$ denotes a single cell. 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. Moreover, we store the height coordinate and different height properties such as, maximum $z_{\max}(s)$, minimum $z_{\min}(s)$ and average $\hat{z}(s)$ of the elevation values within cell s, which quantities will be used later in point cloud segmentation.

$\diamond$ *Hierarchical Grid Model:* Our key idea is to create an extended grid based approach (see Fig. 2) called *hierarchical grid model* which uses a coarse and dense

grid resolution. The cell s of the coarse grid level is subdivided into smaller cells $s'_d | d \in \{1, 2, \ldots, \xi^2\}$, with cell side length $W_{s'_d} = W_s/\xi$, where ξ is a scaling factor (used $\xi = 3$). We store each 3-D point in the coarse and dense grid level as well. We use this data construction to perform object detection, as detailed in Section 3.2.

3.1   Point cloud segmentation using a simple grid model

In our system, point cloud segmentation is achieved by a *simple grid based* approach. Our goal is to discriminate regions of ground, walls and short street objects in the input cloud. For ground segmentation we apply a locally adaptive terrain modeling approach similarly to [9], which is able to accurately extract the road regions, even if their surfaces are not perfectly planar.
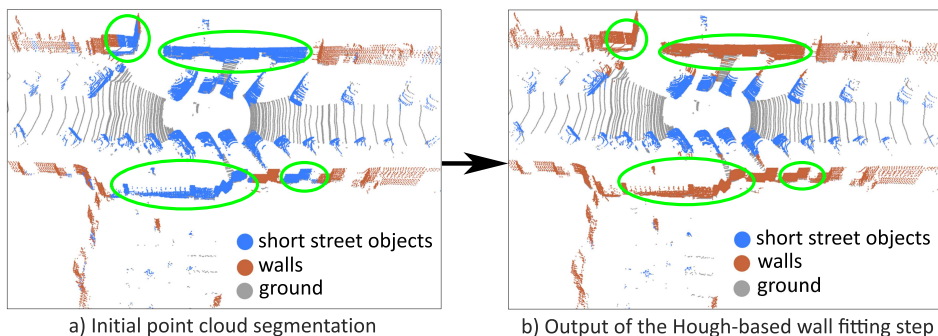

a) Initial point cloud segmentation   b) Output of the Hough-based wall fitting step

Fig. 3. The refinement of the point cloud segmentation result with probabilistic Hough transformation - *(left)* the misclassified cloud regions denoted by green circles. *(right)* Point cloud segmentation after Hough-based wall fitting step.

We use point height information for assigning each grid cell to the corresponding cell class. Before that, we detect and remove grid cells that belong to clutter regions, thus we will not visit these cells later and save processing time. We classify any cell to *clutter*, which contains less points than a predefined threshold (typically 4-8 points). After clutter removal all the points in a cell are classified as *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 object*s (e.g. traffic signs, building walls, lamp post etc.), 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* belonging to vehicles, pedestrians, mail boxes, billboards etc. Due to the limited vertical view angle of the Velodyne LIDAR ($+2°$ up to

-24.8° down), the defined elevation criteria may fail near to the sensor position. In narrow streets where road sides located closely to the measurement position, several nearby grid cells can be misclassified regularly *e.g.* some parts of the walls and the building facades are classified to *short street object* cell class instead of *tall object* cell class (see Fig. 3a)). Our aim is to filter out all of the tall objects, facades and wall structures from the scene, and use only the *short object* class labels for object detection. For this purpose we proposed a probabilistic Hough transformation based segmentation refinement. The grid cells with class labels *tall object* and *short street object* were projected into a pixel lattice (*i.e.* an image), and a probabilistic Hough transformation [12] was used to detect long - elongated structures, which belong to facades or walls in the original point cloud, thereafter the detected lines were back projected into a cloud. The class labels of the grid cells are updated from *short street object* to *tall object* if 1): the grid cell position fits the detected Hough-lines, and 2): the class label of the grid cell was *short street object* before the Hough based refinement step (see Fig. 3b)).

3.2   Urban object detection with a hierarchical grid model

In this section we present the object detection step of our framework. Our aim is to find distinct groups of points which belong to different urban objects on the scene. We used the initial segmentation from Section 3.1, with considering the *short object* cell class as *foreground*, while we label the other classes as *background*. For this task we use the *hierarchical grid model*: On one hand, the coarse grid resolution is appropriate for a rough estimation of the 3-D blobs in the scene, in this way we can also roughly estimate the size and the location of possible object candidates. In addition, we optimize the grid resolution parameter with a statistical approach (see Section 4), instead of setting the cell size parameters by hand similarly to [7], [8]. On the other hand, using a dense grid resolution beside a coarse grid level, is efficient for calculating point cloud features from a smaller subvolume of space, therefore we can refine the detection result derived from the coarse grid resolution. The proposed object detection algorithm consists of three main steps: *First*, we visit every cell of the coarse grid and for each cell s we consider the cells in its 3×3 neighborhood. We visit the neighbor cells one after the other in order to calculate two different point cloud features: (i) the maximal elevation value $Z_{max}(s)$ within a coarse grid cell and (ii) the point cloud density (*i.e.* point cardinality) of a dense grid cell. *Second* our intention is to find connected 3-D blobs within the foreground regions, by merging the coarse level grid cells together. We use an elevation-based cell merging criterion to perform this step. $\psi(s, s_r) = |Z_{max}(s) - Z_{max}(s_r)|$ is a merging indicator, which measures the difference between the maximal point elevation within cell s and its neighboring cell s_r. If the ψ indicator is smaller than a predefined value, we assume that s and s_r belong to the same 3-D object. *Third*, we perform a detection refinement step on the dense grid level. The elevation based cell merging criterion on the coarse grid level often yields that nearby and self-occluded objects are merged into a same blob. We handle this issue by measuring the point density in each sub-cell s'_d at the dense grid level. Our assumption is here that the nearby objects, which

were erroneously merged at the coarse level, could be appropriately separated at the fine level, as the examples in Fig. 4 show. Note that using our Velodyne Lidar camera, the density of the recorded point cloud strongly decreases as a function of the distance from the sensor. We had to compensate this effect by a sensor distance based weighting of the cells during the density based merging step. After the weighting step, we expect by an order of magnitude similar point density in each sub-cell s'_d which belongs to the object candidates. On the other hand, if we observe several empty or low-density sub-cells on the border of two neighboring super-cells, or in the center line of a large cell we can assume that the blob extracted at the coarse level should be divided into two objects. Let us present three typical urban scenarios when the *simple* coarse grid model merges the close objects to the same extracted component, while using a *hierarchical* grid model with coarse and dense grid level, the objects can be appropriately separated. We consider two neighboring super-cell pairs -marked by red - in Fig. 4a) and Fig. 4b), respectively. In both cases the cells contain points from different objects, which fact cannot be justified at the coarse cell level. However, at the dense level, we can identify connected regions of near-empty sub-cells (denoted by gray), which separate the two objects. Fig. 4c) demonstrates a third configuration, where a super-cell intersects two objects, but at the sub-cell level, we can find a separator line.
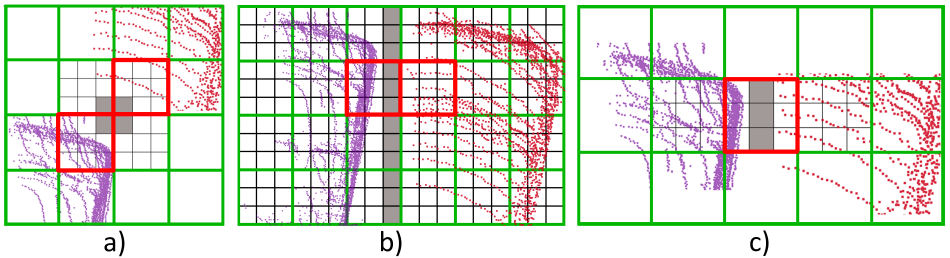

Fig. 4. Separation of close objects at the dense grid level. [color codes: green lines =coarse grid level, black lines=dense grid level, grey cells= examined regions for object separation]

4   Data Charasteristic Analysis and Parameter Sensitivity

Data Characteristic Analysis:
    By using a terrestrial laser scanner, such as the Velodyne LIDAR the data density decreases significantly as function of measurement distance from the sensor. This inhomogeneous point density makes the cell-merging based object detection task challenging. In order to compensate these artifacts for our sensor, we analyzed 1600 relevant frames from three different urban scenarios containing main roads, narrow streets and intersections. We create rings around the sensor
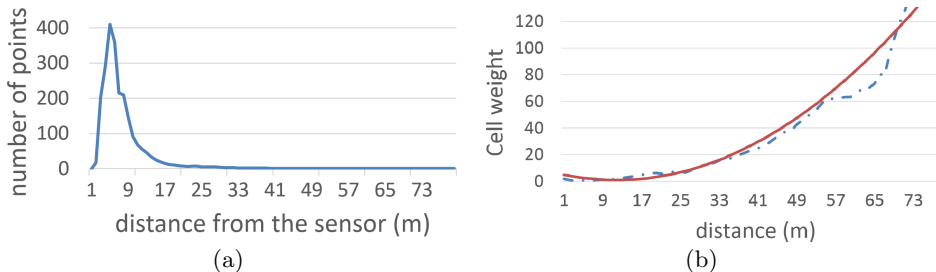
Fig. 5. (a) Point density vs. measurement distance from the sensor. (b) Grid cell weights vs. measurement distance from the sensor. [Note: color codes of Fig. (b): blue = derived weight function, red= sixth-degree polynomial fit of the weight function]

position, thereafter we set the width of each ring to 1m, and we shift the disjunct rings from 1 to 80 meter from the sensor. Finally we measure the distribution of the point density in every ring normalized by the ring area as shown in Fig. 5a). We derive a weight distribution by normalizing the point density function with the maximal point density, and use this function for create weights for the coarse and dense grid cells of the *hierarchical grid model*. Near to the sensor the weight distribution does not modify the point density of the cell, while far from the sensor where the grid cells might contain less points, we enrich the point density by the sixth-degree polynomial fit to the weight distribution, as shown in Fig. 5b).
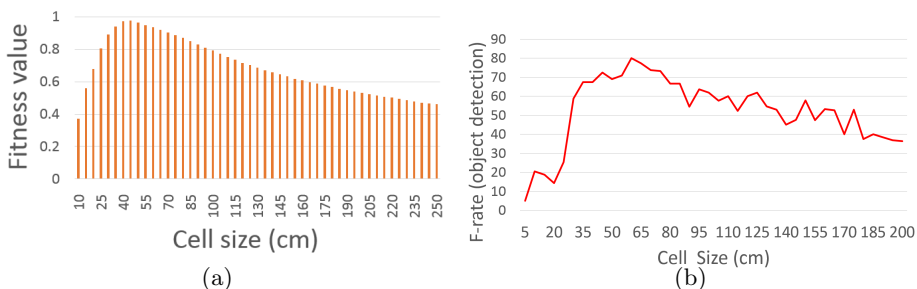
Parameter Sensitivity:

Fig. 6. (a) The distribution of the proposed *cell fitness value* for estimating optimal grid resolution. (b) The F-rate values (harmonic mean of precision and recall) of the detection step as a function of cell size.

In case of a grid based detection task one of the major factors, which affect the accuracy and the speed of the algorithm is the grid resolution (*i.e.* cell size). In order to approximate the optimal range of grid resolution, we propose

a statistical metric called *cell fitness value*, which measures the ratio of *dense* (*d*), *sparse* (*s*) and *empty* (*e*) grid cells in different grid resolutions. We call a grid cell *dense* if it is contains more point than a minimal threshold $t_{(min)}$. We experienced that our initial point cloud segmentation method needs at least 20 points in a cell for appropriate results, therefore we choose $t_{(min)} = 20$. Finally we derived the *cell fitness value* $f \in [0, 1]$ as follows: $f = \frac{\#d}{(\#d + \#s) - \#e}$, where $\#$ denotes the number of the cells on the screen (see Fig. 6a)), in order to maximize the relative frequency of the *dense* grid cells. Moreover, the distribution of the *cell fitness value* f clearly has a maximum range as a function of grid resolution, therefore we choose an optimal grid resolution corresponds to this maximum range (used 60 cm).

5   Performance Evaluation and Conclusion

Point Cloud Dataset	NO	Conn. Comp. Analysis [14]		Prop. Hierarchical Grid	
		F-rate(%)	Avg. Processing Speed (fps)	F-rate(%)	Avg. Processing Speed (fps)
Budapest Dataset #1	669	77	0.38	89	29
Budapest Dataset #2	429	64	0.22	79	25
KITTI Dataset [5]	496	75	0.46	82	29
Overall	1594	72	0.35	83	28

Table 1. Numerical comparison of the detection results obtained by the Connected Component Analysis [14] and the proposed *Hierarchal Grid Model*. The number of objects (NO) are listed for each data set, also and in aggregate.

We evaluated our method in three urban LIDAR sequences, concerning different urban scenarios, such as main roads, narrow streets and intersections. Two scenarios recorded in the streets of Budapest, one scenario has been selected from the KITTI Vision Benchmark Suite [5]. The data flows have been recorded by a Velodyne HDL-64E S2 camera with a 10Hz rotation speed. We have compared our *hierarchical grid model* to a connected component analysis which uses kd-tree based solution for range search [14]. Qualitative results on four sample frames are shown in Fig. 7 and in Fig. 8.[1] For Ground Truth (GT) generation, we have developed a 3-D annotation tool, which enables labeling the urban objects manually as object or background. We manually annotated 1594 urban objects. To enable fully automated evaluation, we need to make first a non-ambiguous assignment between the detected objects and ground truth (GT) object samples. We use Hungarian algorithm [10] to find maximum matching. Thereafter, we counting the Missing Objects (MO), and the Falsely detected Objects (FO). These values are compared to the Number of real Objects (NO), and

---

[1] Demonstration videos and GT data are also available at the following url:
   http://web.eee.sztaki.hu/~borcs/demos.html

the F-rate of the detection (harmonic mean of precision and recall) is also calculated. We have also measured the processing speed in frames per seconds (fps). The numerical performance analysis is given in Table 1. The results confirms that proposed model surpasses the Connected Component Analysis technique in F-rate for all the scenes. Moreover, the proposed *Hierarchical Grid Model* significantly faster on streaming data, and less influenced by the inhomogeneous density of the point cloud. In urban point clouds we measure 0.35 fps average average processing with Connected Component Analysis [14] and 28 fps with the proposed *Hierarchical Grid Model.*
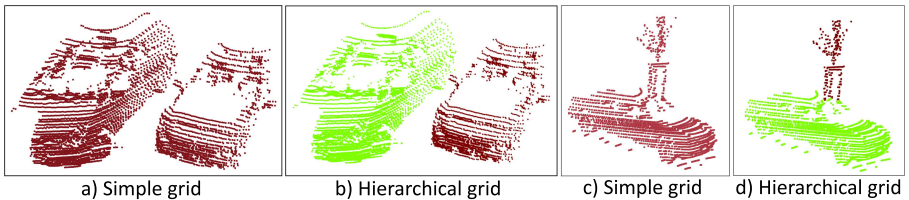

a) Simple grid   b) Hierarchical grid   c) Simple grid   d) Hierarchical grid

Fig. 7. Object separation for a case of nearby objects. Comparison of the *Simple Grid Model* Fig. a), c) and the *Hierarchical Grid Model* Fig. b), d).
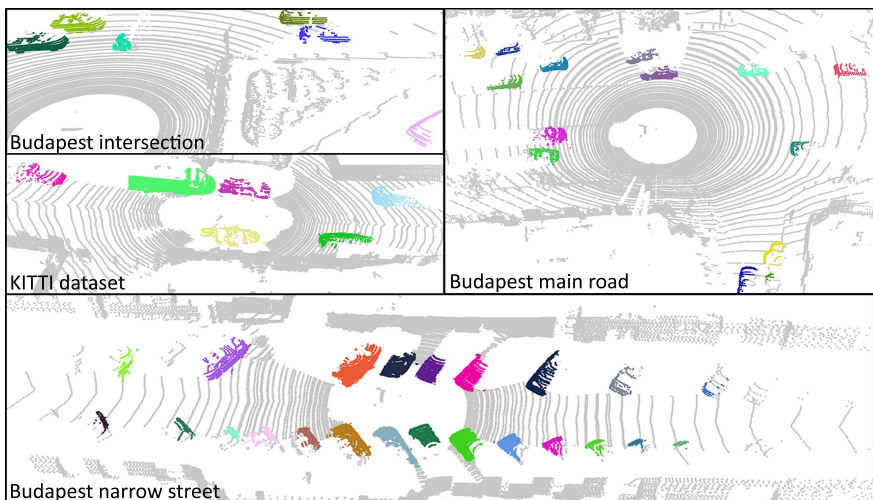

Fig. 8. Object detection results on different urban scenarios.

As a conclusion, we have proposed a novel data structure, called *Hierarchical Grid Model* and corresponding connected component analysis algorithm to find distinct groups of 3-D points which belong to different urban objects in LIDAR

point clouds. We propose a statistical metric for approximation of optimal grid resolution in terms of object detection. The model has been quantitatively validated based on Ground Truth data, and the advantages of the proposed solution versus a baseline technique have been demonstrated.

References

1. Azim, A., Aycard, O.: Detection, classification and tracking of moving objects in a 3D environment. In: Intelligent Vehicles Symposium. pp. 802–807 (2012)
2. Behley, J., Steinhage, V., Cremers, A.B.: Performance of histogram descriptors for the classification of 3D laser range data in urban environments. In: ICRA. pp. 4391–4398. IEEE
3. Benedek, C., Molnár, D., Szirányi, T.: A Dynamic MRF Model for Foreground Detection on Range Data Sequences of Rotating Multi-Beam Lidar. In: International Workshop on Depth Image Analysis, LNCS. Tsukuba City, Japan (2012)
4. Frederickson, G.N.: Data structures for on-line updating of minimum spanning trees. In: Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing. pp. 252–257. STOC '83, ACM, New York, NY, USA (1983), http://doi.acm.org/10.1145/800061.808754
5. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
6. Golovinskiy, A., Funkhouser, T.: Min-cut based segmentation of point clouds. In: IEEE Workshop on Search in 3D and Video (S3DV) at ICCV (Sep 2009)
7. Himmelsbach, M., Müller, A., Lüttel, T., Wünsche, H.J.: LIDAR-based 3D Object Perception. In: Proceedings of 1st International Workshop on Cognition for Technical Systems. München (Oct 2008)
8. Hu, H., Munoz, D., Bagnell, J.A., Hebert, M.: Efficient 3-d scene analysis from streaming data. In: IEEE International Conference on Robotics and Automation (ICRA) (2013)
9. Józsa, O., Börcs, A., Benedek, C.: Towards 4D virtual city reconstruction from Lidar point cloud sequences. In: ISPRS Workshop on 3D Virtual City Modeling, ISPRS Annals Photogram. Rem. Sens. and Spat. Inf. Sci., vol. II-3/W1, pp. 15–20. Regina, Canada (2013)
10. Kuhn, H.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly 2, 83–97 (1955)
11. Lalonde, J.F., Vandapel, N., Hebert, M.: Data structures for efficient dynamic processing in 3-d. The International Journal of Robotics Research 26(8), 777–796 (August 2007)
12. Matas, J., Galambos, C., Kittler, J.: Robust detection of lines using the progressive probabilistic Hough transform. Comput. Vis. Image Underst. 78(1), 119–137 (Apr 2000), http://dx.doi.org/10.1006/cviu.1999.0831
13. McNaughton, M., Urmson, C., Dolan, J.M., Lee, J.W.: Motion planning for autonomous driving with a conformal spatiotemporal lattice. In: ICRA. pp. 4889–4895. IEEE (2011)
14. Rusu, R.B., Cousins, S.: 3D is here: Point cloud library (pcl). In: International Conference on Robotics and Automation. Shanghai, China (2011)
15. Thrun, S., et. al.: Stanley: The robot that won the darpa grand challenge. Journal of Field Robotics 23(9), 661–692 (June 2006)