Segmentation of 3D Lidar Data in non-flat Urban Environments using a Local Convexity Criterion

Frank Moosmann, Oliver Pink and Christoph Stiller
Institut für Mess- und Regelungstechnik
Universität Karlsruhe (TH), 76128 Karlsruhe, Germany
Email: {moosmann,pink}@mrt.uka.de

Abstract—**Present object detection methods working on 3D range data are so far either optimized for unstructured offroad environments or flat urban environments. We present a fast algorithm able to deal with tremendous amounts of 3D Lidar measurements. It uses a graph-based approach to segment ground and objects from 3D lidar scans using a novel unified, generic criterion based on local convexity measures. Experiments show good results in urban environments including smoothly bended road surfaces.**

I. INTRODUCTION

One of the key tasks intelligent vehicles have to perform is the reliable perception of their environment, namely the detection of obstacles and free space. Especially for the latter, laser scanners have proven efficient, as their resolution and field of view exceeds radar and ultrasonic sensors and they provide direct distance measurements. Most reported in literature are sensors that conduct 2D scanning ([1], [2] and many others), *i.e.* scanning is performed along a plane within a limited viewing angle. Mounted parallel to the ground plane, each scan acquires a 1D sequence of range and angle measurements. This allows an easy detection of obstacles (and implicitly free-space) by applying 1D signal processing methods. However, objects above or below the scanning plane cannot be detected. The limited number of measurements thereby complicates classification and tracking of obstacles. Additionally, in hilly environments the scanner might fail to detect obstacles or might recognize an ascending road as obstacle.

Within the last few years, fully three-dimensional scanners have been introduced. Rather than scanning in a plane, 3D volumes are scanned resulting in a (potentially unordered) cloud of 3D points. This allows detection of all kinds of obstacles and the explicit detection of free-space as the ground is also sampled. However, the vast amount of data poses a great challenge on the algorithms. Furthermore, the multidimensional signal necessitates multidimensional signal processing methods.

We propose a method that is capable of handling this data. The major contributions are: First, we introduce a novel criterion dubbed *Local Convexity* that uses local geometric features. It can be generically used to segment both ground and objects. Second, we present a segmentation algorithm that keeps the full 3D information delivered by the sensor unlike many popular approaches. We show that the combination works successfully even in non-flat urban environments.

This paper is organized as follows. In the next section we first give an outline of relevant work, followed by the detailed description of our approach. In section IV we provide experimental results. Section V concludes this paper and gives an outlook for future research in this area.

II. PREVIOUS WORK

Object detection algorithms are often classified into two groups: Model-based and model-free. Model-based approaches try to solve detection and classification simultaneously by fitting models to the data, as *e.g.* [3], [4]. This, however, is computationally demanding and hardly applicable in real-time.

The alternative and more flexible approach is to first separate obstacles from the ground using a generic model. To reduce processing time, a dimensionality reduction is usually applied. According to the type of projection, the approaches can basically be divided into two subclasses: Projection to a ground plane and projection to a virtual image plane resulting in a so-called *range image*.

A. Ground Plane based methods

Most frequently, data is projected to an assumed or estimated ground plane, often combined with an occupancy grid map [5]. One of the advantages is that several sensors can be fused easily (even in probabilistic ways [6]), and that mapping is straight-forward. Many teams participating in the DARPA Urban Challenge, including ours, successfully applied this method (e.g. [7], [8], [9]). However, the difficulty of distinguishing between obstacles and ground still remains. A simple solution is to define the density of the points within a cell as occupancy value [9]. This easily marks all cells containing vertical structures as occupied. Object extraction is then usually performed by clustering connected occupied cells together, as *e.g.* in [10]. While these approaches work very well for environments with a level ground plane and vertical wall structures, they are not suitable for reliable detection of sloped objects like vegetation, hills, or curbs as they often appear in outdoor environments. Furthermore, a substantial portion of the information available is lost due to the projection. Therefore, these approaches are very well suited to obstacle avoidance in flat areas, but give only very coarse information on object geometry.

B. Range Image based methods

Alternatively, the 3D data can be projected onto a cylinder whose axis is the rotational axis of the scanner. This projection

978-1-4244-3504-3/09/$25.00 ©2009 IEEE

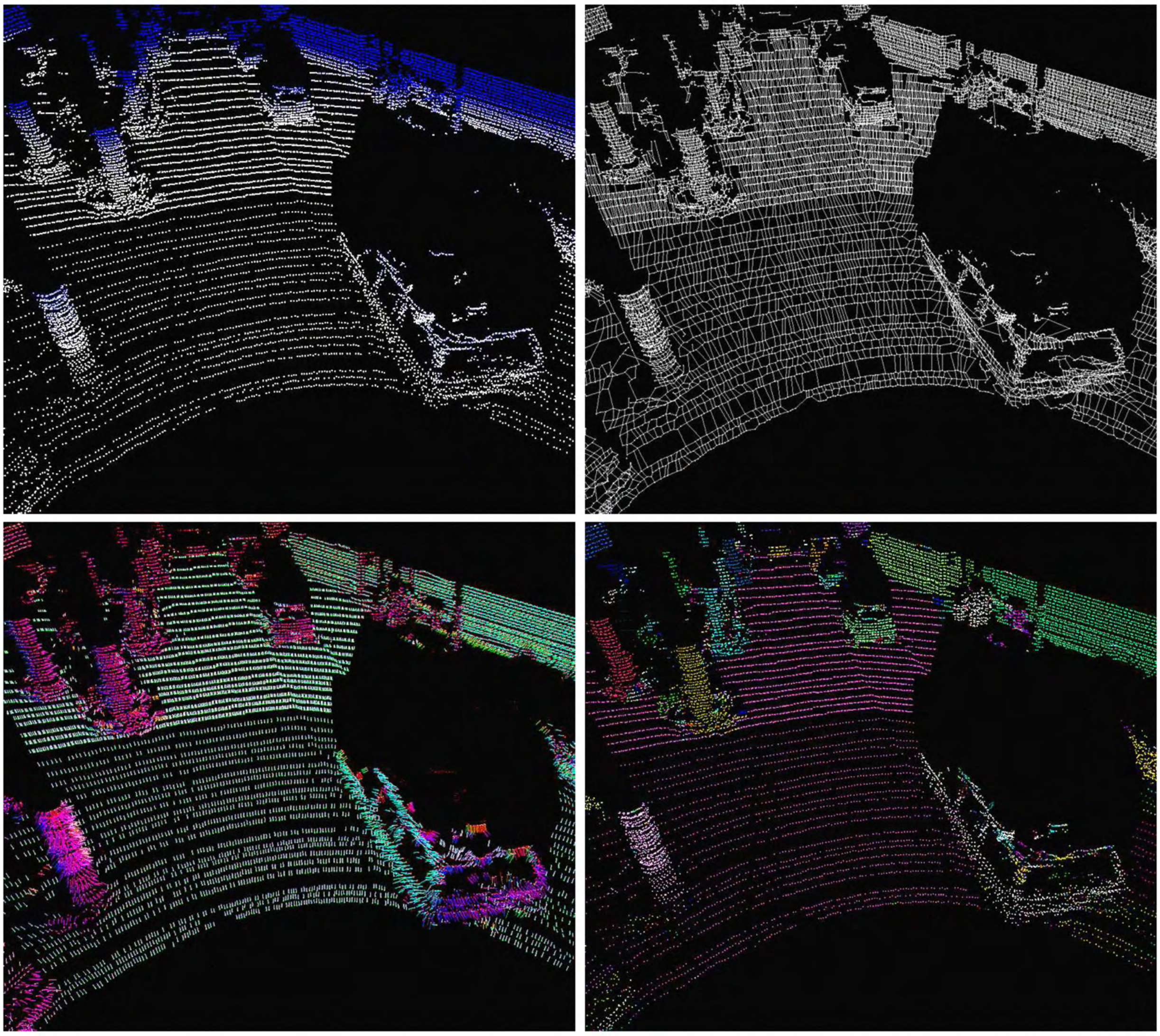

Fig. 1. Illustration of our method, best viewed in color: (top-left) 3D point cloud, blueness decodes height above ground. Street is in vertical direction with parked cars and a wall on the right and tree trunks on the left. (top-right) Neighborhood Graph. (bottom-left) Normal vectors, direction is indicated both by color and a small line. (bottom-right) Segmentation result.

yields a range image, where pixel values correspond to the distance measurements. Founding work has been carried out by Hoover *et al.* [11], whose principal approach of local (planar) surface fitting and clustering afterwards is still followed today, as e.g. in [12]. However, range image based methods are mostly used in controlled, artificial environments, so their use within intelligent vehicles is rare.

More popular is the use of (2-dimensional) graphs, a natural generalization of images. Most popular segmentation methods working on graphs are not surprisingly adaptions of (intensity) image based methods. The first step is to build a weighted graph, where the weights represent the similarity between two connected nodes. The graph is usually obtained directly from the scanning setup or by triangulation. Attributes used include surface direction [13], curvature [14], [15], edges[16], local smoothness constraints [17] and more. Region growing algorithms (*e.g.* in [13], [18]) can be used for segmentation, which are easy to implement and fast. However, different choices of seed points usually result in different segments. Hence, such algorithms are not considered as robust. Graph cuts based algorithms play an important role, like the normalized cuts algorithm of [19], which minimize some cost function. One advantage is the guarantee to converge to the global minimum. A major drawback for automotive applications clearly is their computational cost. Aguiar *et al.* [20] alternatively generate a minimum spanning tree and apply recursive cutting. This keeps nearly linear costs while still finding a good local minimum.

Motivated by our experiences on the ground plane based methods from the DARPA Urban Challenge, we adapt in this work the graph-based approach using the full 3D data. This way, all information included in the multidimensional nature of the signal is exploited. We show that this allows for a fast algorithm and good results even on bended road surfaces.

III. PROPOSED METHOD

Our method is a scan-wise segmentation of the environment which is performed in several steps (see Fig. 1): scan acquisition, neighborhood graph construction, attribute calculation, and finally segmentation. We further introduce an additional classification step to decide about a segment belonging to ground or obstacle. These steps are detailed in the following.

A. Scan Acquisition

For scan acquisition we use a Velodyne HDL-64 High Definition Lidar scanner which is mounted on top of our experimental vehicle. It consists of a column of 64 single lasers, covering a pitch range of approximately 26 degrees. It rotates at a rate of 10 Hz sweeping the complete horizontal ground plane and producing 180000 points per turn. As the pitch angles of the lasers are fixed, each of the lower lasers produces a ring of point measurements on a horizontal plane. The top-left of Fig. 1 depicts an example scan.

In general, our method is not restricted to one specific type of sensor, however, the physical setup of the Velodyne HDL-64 allows to directly obtain a neighborhood graph from a scan, as explained next.

B. Neighborhood Graph Construction

The goal of this step is to turn a scan into an undirected graph $G = \{N, E\}$, with the nodes $N = \{(x, y, z)_i\}$ being the measured points in 3D, and the edges $E = \{(N_i, N_j)\}$ connecting the nodes. As connections we chose an image-like 4-neighborhood, as this can be directly obtained from the scanner. As the scanner turns counter-clockwise, a point's left neighbor corresponds to the next measurement obtained by the same laser, its right neighbor to the previous measurement. The upper and lower edges are assigned to the measurements of the upper and lower laser diode with the most similar yaw angle, respectively.

The graph is further postprocessed by deleting any edge that exceeds a certain absolute distance threshold or that exceeds a distance threshold relative to neighboring connections. The top-right picture of Fig. 1 shows the calculated graph corresponding to the top-left picture.

In case scan acquisition merely delivers an unordered point cloud, a graph could e.g. be obtained by triangulation or other neighborhood search methods. This, of course, will excessively increase processing time.

C. Attribute Calculation

For each of the nodes we calculate an attribute which is used in the next step to segment the graph. The criterion presented in section III-D is a criterion which is based on local surface geometry. As range sensors sample surfaces only pointwise, the surface geometry has to be interpolated from the range data. In this work, we approximate the local surface around a point by a plane or rather its normal vector. A robust method would be to use several neighboring points and utilize least squares estimation, however, this is not applicable in real-time. Alternatively, as in [13], we calculate the cross product of the displacement vectors between the left and lower, lower and right, right and upper and finally upper and left vectors (in case the links still exist in the graph). We geometrically average the obtained vectors to obtain the estimated local surface normal. In order to reduce noise, a moving average filter is applied to the field of surface normals. Results are illustrated in the bottom-left picture of Fig. 1.

D. Segmentation

Based upon the observation that many object parts have convex outlines and that a vertical structure usually represents one single object, we introduce a combined segmentation criterion in the following. Two points (*i.e.* nodes) belong to the same object, if and only if there exists a path in the graph connnecting the two nodes, with every edge of that path fulfilling this criterion.

The criterion is dependent on two neighboring surfaces s_i and s_j, characterized by their point locations $\vec{p}_{i/j}$, respective to an arbitrary but fixed coordinate system, and their normals $\vec{n}_{i/j}$, with $||\vec{n}_i|| = ||\vec{n}_j|| = 1$. Normal vectors are assumed to point outwards.

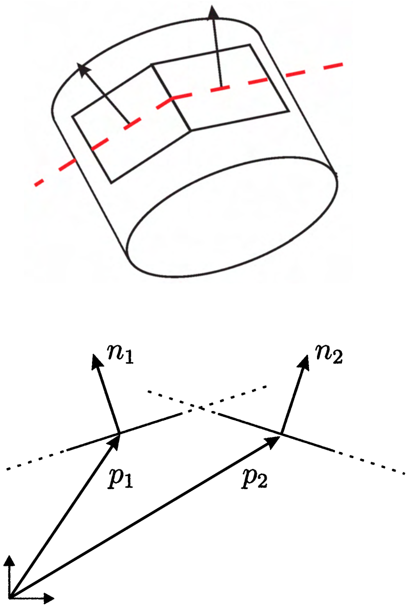

Fig. 2. Local convexity holds if the center point of a surface is below the other surface and vice versa.

A core concept for object segmentation proposed in this contribution is the following definition of *Local Convexity*. As illustrated in Fig. 2, two neighboring surfaces s_i and s_j are termed *locally convex* to each other, if the center point $\vec{p_i}$ lies below the surface s_j and vice versa. In order to increase robustness, we also consider two surfaces s_i and s_j as *locally convex*, if their normal vectors approximately have the same direction. So in contrast to the common convexity definition as a curve property, *Local Convexity* refers to a *local, pairwise* relation.

This is a criterion that is very fast to evaluate by using the dot product. The following equation expresses whether two

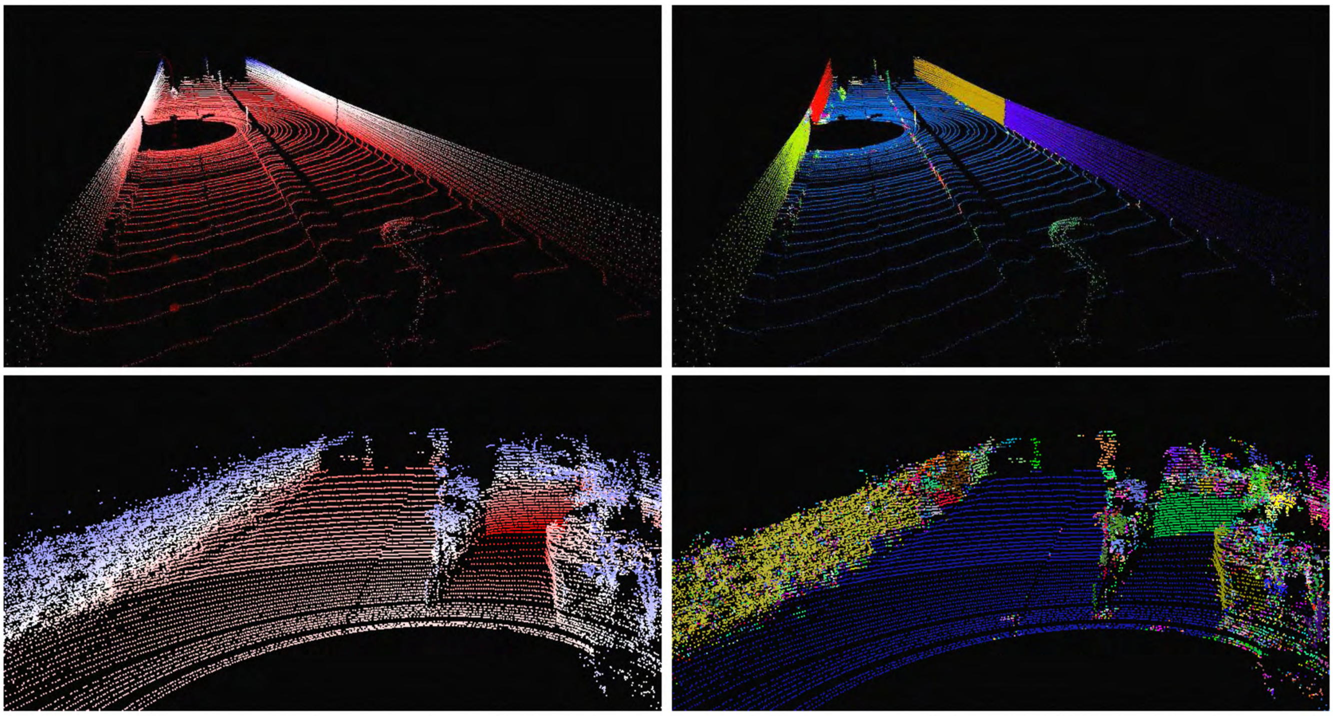

Fig. 3. Segmentation results on non-flat areas, best viewed in color. (first row) Concave street in a tunnel. (second row) Convex gateway to an underground garage. (left column) 3D point cloud. Blue decodes height above ground, red decodes depth below ground. (right column) Segments, displayed in different colors.

surfaces s_i and s_j are *locally convex* to each other:

$$\mathrm{lc}(s_i, s_j) = \begin{cases} \text{true} & [(\vec{n}_i \cdot \vec{d}_{ij} \leq ||\vec{d}_{ij}|| \cdot \cos(\frac{\pi}{2} - \epsilon_1)) \\ & \wedge\ (\vec{n}_j \cdot \vec{d}_{ji} \leq ||\vec{d}_{ji}|| \cdot \cos(\frac{\pi}{2} - \epsilon_1)))] \\ & \vee\ [\vec{n}_i \cdot \vec{n}_j \geq 1 - ||\vec{d}_{ij}|| \cdot \cos(\frac{\pi}{2} - \epsilon_2)] \\ \text{false} & \text{else} \end{cases}$$

with $\vec{d}_{ij} = (\vec{p}_j - \vec{p}_i)$ being the displacement vector from $\vec{p}_i$ to $\vec{p}_j$ and hence $\vec{d}_{ji} = -\vec{d}_{ij}$. The parameter ϵ_1 can be chosen to set the level of concavity (zero = none) as it defines the angle in which the other point may lie above the surface plane. $\epsilon_2 \geq 0$ defines the distance-adaptive similarity angle and can be chosen to account for noisy data, as mentioned above.

Finally, we combine *Local Convexity* with a vertical structure criterion to obtain the segmentation criterion

$$\mathrm{grow}(s_i, s_j) = \mathrm{lc}(s_i, s_j) \vee [(|z_{\vec{n}_i}| < \epsilon_3) \wedge (|z_{\vec{n}_j}| < \epsilon_3)].$$

Here, $0 \leq \epsilon_3 \leq 1$ is a third constant that can be chosen to connect two vertical surfaces. By setting it to 1 only one global segment would be obtained, as the absolute value of a normal vector's z coordinate is always smaller or equal to 1.

Thus, two surfaces are combined into a segment, if they are either *locally convex*, or both normal vectors are approximately vertical. One important property follows from this definition: If $\mathrm{grow}(s_i, s_j)$ holds, then $\mathrm{grow}(s_j, s_i)$ holds as well, *i.e.* the criterion is symmetric. This allows the segmentation to be efficiently solved by a region growing algorithm as follows:

Given the attributed graph, the following region growing algorithm is executed until no more surfaces are left:

1) Select a seed node randomly
2) Grow the segment until no more nodes are added
3) Delete the segment from the graph

Even though seed surfaces are selected randomly, the result of the algorithm is deterministic. The growing criterion will merge two connected nodes into the same segment no matter from which direction the connection in the graph is approached.

The segmentation result of our running example is shown on the bottom-right of Fig. 1.

E. Classification

For a first mapping application we further developed a simple classifier to decide about a segment being ground or obstacle. For each segment a histogram over all the normal vector's z value is generated and classified as ground if the topmost bin contains most votes.

IV. EXPERIMENTS

We have evaluated the proposed algorithm for numerous scans acquired by the HDL-64 sensor mounted on a Volkswagen Passat experimental vehicle in inner city traffic scenes. As no ground truth information is available, a qualitative performance evaluation is conducted. The parameters were fixed to $\epsilon_1 = 0.1 \approx 6°$, $\epsilon_2 = 0{,}03 \approx {}^{2°}/{m}$, and $\epsilon_3 = 0.5$ throughout all experiments.

Two example results of the segmentation algorithm can be seen in Fig. 3. Shown are two scenes including both uneven road surfaces (one slightly concave, the other convex). Our

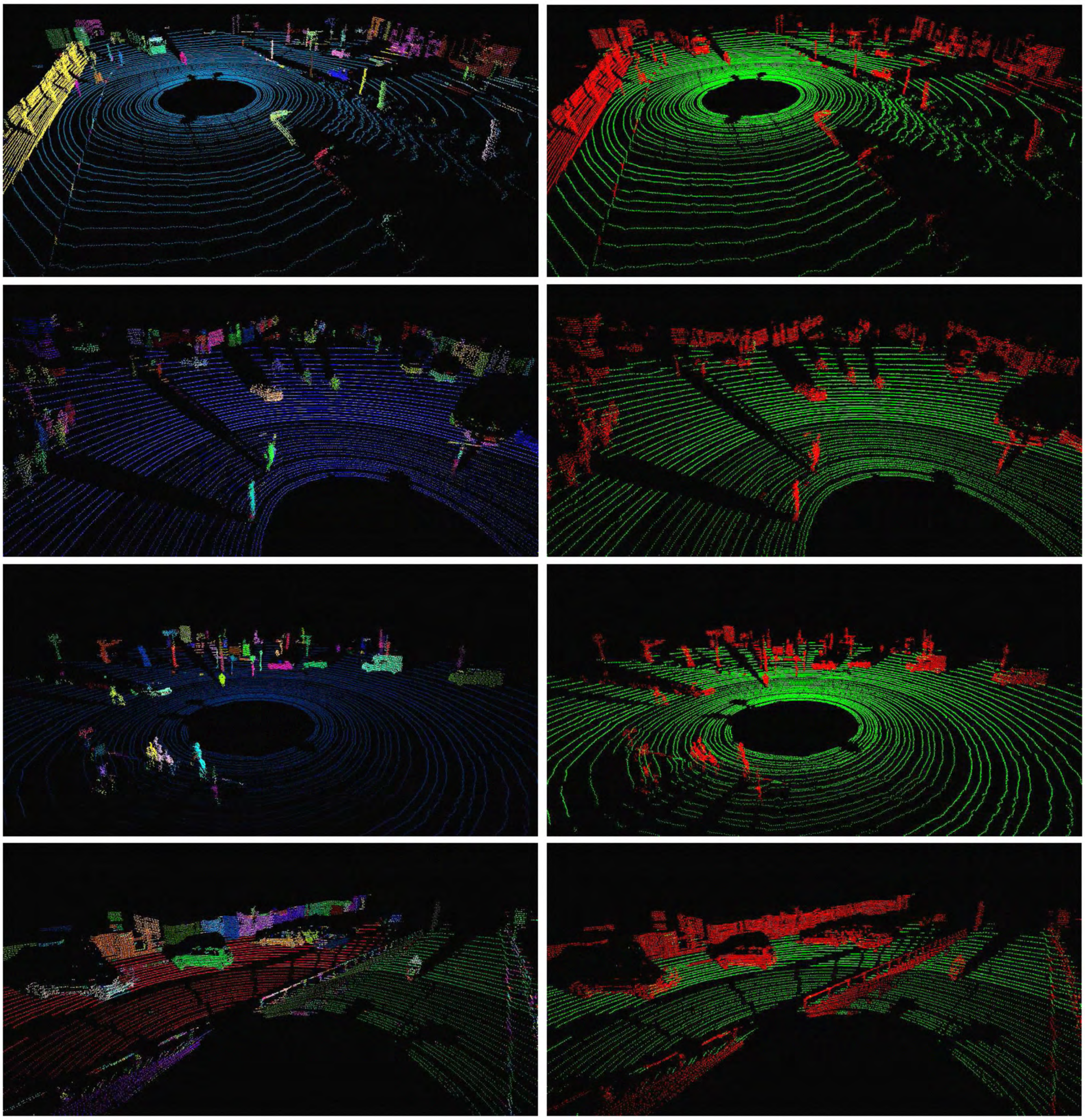

Fig. 4. Segmentation and classification results, best viewed in color. (first row) Broad road with pedestrians, a motorcyclist, a truck, cars and tree trunks. (second row) Another intersection with pedestrians and cars. (third row) Intersection with pedestrians, cyclists, and cars. (forth row) Entry to an underground passage with motorcyclist. Cars parking and driving on upper level. (left column) Segments, displayed in different colors. (right column) Classification results, green indicating ground, red indicating obstacle.

algorithm is able to successfully merge the road segment. The walls in the upper picture are split into two segments due to shadows breaking the connectivity. In the lower picture even the vegetation on the side of the horizontal street is merged into one (yellow) segment. However, vegetation is in general very difficult to deal with, as range measurements are extremely noisy thereon. This can be seen on the right, where many one-point-segments were formed.

Further results including the classification outcomes are depicted in Fig. 4. As visible in the left column, usually all traffic participants are clearly segmented from the ground (best illustrated by the first three rows). However, we sometimes encounter the typical problems that most segmentation algorithms have: both over- and undersegmentation can occur, with oversegmentation being the most common but also the less critical. Running on a sequence, however, merged segments are usually split in some frames, leaving it to a – so far missing – tracking algorithm to merge segments by detecting uniformly moving ones. The fourth row was again taken from uneven terrain, with the green segment being a street descending into a tunnel, the red segment a street on the upper ground. Nevertheless, the segmentation result is very good. The column on the right illustrates the classification results.

We ran the computations on a Pentium M with 2.1GHz using unoptimized code. The average processing times per frame are 0.352 seconds for normal calculation, 0.25 seconds for segmentation and $0.021s$ for classification. With a scanning frequency of 10Hz, this is obviously not real-time. However, we are very confident that real-time performance is possible by optimizing and parallelizing code and running it on most recent hardware.

V. CONCLUSIONS AND FUTURE WORK

We presented a novel segmentation algorithm that is capable of segmenting objects from the ground even in non-flat environments. While data projection methods inherently loose information along the dimension of projection, the proposed method considers the physical setup of the scanner to turn the data into a 2D graph, not discarding important information but allowing for efficient processing. The concept of *Local Convexity* represents the core of the algorithm. It has been introduced as a novel geometric criterion allowing for a generic segmentation method that is at the same time capable of segmenting non-flat ground and common obstacles. We demonstrated that the proposed algorithm achieves good results on data acquired in urban environments.

Future work will consist of augmenting the discriminating properties of our method by a classification framework such as [10]. Further steps include the application of temporal tracking. It would be interesting to develop a tracking method that makes use of appearance-learning in full 3D, extending the 2D method of Gate *et al.* [2].

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the German collaborative research center on Cognitive Automobiles (SFB/Tr28) granted by Deutsche Forschungsgemeinschaft.

REFERENCES

[1] F. Nashashibi and A. Bargeton, "Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation," *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 847–852, June 2008.

[2] G. Gate and F. Nashashibi, "Using targets appearance to improve pedestrian classification with a laser scanner," *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 571–576, June 2008.

[3] T. Rabbani and F. van den Heuvel, "Efficient hough transform for automatic detection of cylinders in point clouds."

[4] S. J. Ahn, I. Effenberger, S. Roth-Koch, and E. Westkämper, "Geometric segmentation and object recognition in unordered and incomplete point cloud," in *DAGM-Symposium*, 2003, pp. 450–457.

[5] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Auton. Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[6] M. Tay, "An efficient formulation of the Bayesian occupation filter for target tracking in dynamic environments," *International Journal of Vehicle Autonomous Systems*, vol. 6, pp. 155–171(17), 31 December 2007.

[7] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The Stanford entry in the Urban Challenge," in *Journal of Field Robotics*, ser. 9, vol. 25, September 2008, pp. 569 - 597.

[8] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the Urban Challenge," in *Journal of Field Robotics*, vol. 25, no. 8, 2008, pp. 425 - 466.

[9] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. von Hundelshausen, O. Pink, C. Frese, and C. Stiller, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," in *Journal of Field Robotics*, ser. 9, vol. 25, 2008, pp. 615 - 639.

[10] 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.

[11] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher, "An experimental comparison of range image segmentation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 7, pp. 673–689, 1996.

[12] J. M. Biosca and J. L. Lerma, "Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods," in *ISPRS Journal of Photogrammetry & Remote Sensing*, vol. 63, 2008, pp. 84 - 98.

[13] L. Spinello, R. Triebel, and R. Siegwart, "Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction," *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 1823–1829, Sept. 2008.

[14] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 2, pp. 167–192, 1988.

[15] A. Jagannathan and E. L. Miller, "Three-dimensional surface mesh segmentation using curvedness-based region growing approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2195–2204, 2007.

[16] D. Steinhauser, O. Ruepp, and D. Burschka, "Motion segmentation and scene classification from 3D LIDAR data," *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 398–403, June 2008.

[17] T. Rabbani, F. van den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," 2006, pp. xx–yy.

[18] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3D laser data," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 4043–4048, May 2008.

[19] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.

[20] C. S. R. Aguiar, S. Druon, and A. Crosnier, "3D datasets segmentation based on local attribute variation," in *Intelligent Robots and Systems (IROS) 2007. IEEE/RSJ International Conference on*, October 2007, pp. 3205 - 3210.