Obstacle Prediction for Automated Guided Vehicles Based on Point Clouds Measured by a Tilted LIDAR Sensor

Zoltan Rozsa, *Member, IEEE*, and Tamas Sziranyi, *Senior Member, IEEE*

***Abstract*— Environment analysis of automatic vehicles needs the detection from 3-D point cloud information. This paper addresses this task when only partial scanning data are available. Our method develops the detection capabilities of autonomous vehicles equipped with 3-D range sensors for navigation purposes. In industrial practice, the safety scanners of automated guided vehicles (AGVs) and a localization technology provide an additional possibility to gain 3-D point clouds from planar contour points or low vertical resolution. Based on this data and a suitable evaluation algorithm, intelligence of vehicles can be significantly increased without the need for installation of additional sensors. In this paper, we propose a solution for an obstacle categorization problem for partial point clouds without shape modeling. The approach is tested for a known database, as well as for real-life scenarios. In case of AGVs, real-time run is provided by on-board computers of usual complexity.**

***Index Terms*— LIDAR, point cloud, object recognition, autonomous vehicle, automated guided vehicles, keypoint detection, bag of features.**

I. Introduction

INTELLIGENT vehicles are usually provided with data from OSH (Occupational safety and health) devices or narrow Field of View (FoV) LIDAR detectors (e.g., Velodyne VLP-16).[1] 2D and 3D LIDARs with narrow FoV acquire low vertical information about the near environment by one frame. Incremental registration offer a chance for the exploitation of this data type.

Autonomous vehicles/mobile machines are in the process of intensive development including sensors and algorithms. In industrial transportation systems so called AGVs are needed to differentiate obstacle categories so that the gained information can be utilized for many aims: recognized objects can be used as landmarks for navigation purposes or for better respond to a certain safety situation.

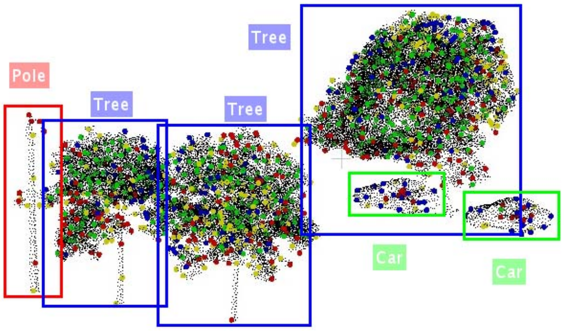


Fig. 1. Point clouds of objects with keypoints clustered on local descriptors.

Autonomous vehicles must be equipped with protective devices because of OSH aspects, implementing different sensor modalities and partly independent, or contrary, fused evaluation of sensor information to get a higher reliability. There are various collision avoidance systems, see the survey in [1]. In case of AGVs, the protective devices are usually safety laser scanners, and one or more regular 2D laser scanners are installed. Conventional use of these sensors can result in go forward, stop or avoid commands for the autonomous vehicle, when an obstacle is detected. These decisions are based on the distance and the static or dynamic nature of the obstacle (differentiation of static and dynamic obstacles is not an easy task itself [2]). A system which is capable of obstacle recognition could suggest avoidance direction (using yet invisible but known extension of the obstacle), and have more precise static/dynamic differentiation; e.g. standing human will not be categorized as static. Calculating given parameters of the partially visible objects (size, maximum acceleration, maximum velocity, etc) can be realized as well. Even prediction of the behavior of the obstacles (vehicles, humans or animals will react differently to the approach of a mobile machine) is a possibility.

The vehicle needs to be informed about the objects in its surroundings. The visible surfaces of these objects are most often represented in a 3D coordinate system as a set of measured points which is referred as 3D point cloud. However, object recognition from 3D point clouds is a competitive research area without applicable results for partial views (the present problem).

3D pattern recognition is a challenging issue both in full 3D and 2.5D cases [3], [4]:

Manuscript received July 21, 2016; revised April 26, 2017 and October 27, 2017; accepted December 28, 2017. This work was supported by the Hungarian Scientific Research Fund under Grant OTKA/NKFIH 120499. The Associate Editor for this paper was J. Zhang. *(Corresponding author: Zoltan Rozsa.)*

Z. Rozsa is with the Research Institute for Computer Science and Control (MTA SZTAKI), Hungarian Academy of Sciences, 1111 Budapest, Hungary, and also with the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, 1111 Budapest, Hungary (e-mail: rozsazoltan@sztaki.hu).

T. Sziranyi is with the Research Institute for Computer Science and Control (MTA SZTAKI), Hungarian Academy of Sciences, 1111 Budapest, Hungary.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2790264

[1] http://velodynelidar.com/vlp-16.html

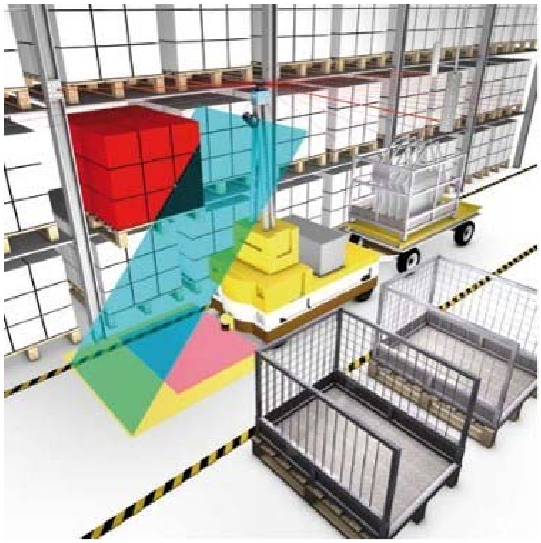


Fig. 2. Tilted sensor installation for overhang detection. Photo source: SICK - Efficient solutions for material transport vehicles in factory and logistics automation.[2]

- State of the art 3D shape recognition methods are capable of e.g. retrieving 3D models from 2D sketch queries applying multi-view convolutional neural networks on rendered models [5].
- Best recognition results on TOSCA+Sumner dataset of 3D Shape Categorization Benchmark is about 96% achieved by 3D Spatial Pyramids [6].
- In Non-rigid 3D Shape Retrieval track of SHREC'15 (Shape retrieval Contest) NN (nearest neighbors) values equal or near to 100 % are obtained [7].

These works make it possible for aiming large-scale object retrieval [8] for researches working with full 3D. While obtaining full 3D model is not possible in real scenarios, recognition from 2.5D can be even more difficult.

Hereinafter we investigate the recognition problem from sequentially acquired information of a tilted LIDAR. This development is devoted for exploiting the on-board sensor of a sensored AGV with improved computing capacity. In case of AGVs, sensors tilted upwards can detect for example hanging crane hooks. Sensors installed tilted downwards can warn about objects jutting out of shelving (Fig. 2). In urban environment typical reason of tilted scanner installation is Mobile Laser Scanning (MLS); illustration of our proposed method on MLS data can be seen in Fig. 1. We explore an object in bottom-up sequence (or top-down) collecting the rare views in front of the vehicle. We would only see the full height 2.5D view of this object if the vehicle will be (dangerously) close to the obstacle or we will not see it at all. It is desirable not to let our machine to approach the object too close, the decision must be made in much earlier stage from partial information. Data of on-board 2D LIDARs have been exploited for better awareness. Using additional sensor information (e.g. camera) is recommended, but each modality should have its own reliability for superior fusion of different on-board solutions.

Our method can solve the recognition problem for wide FoV 3D LIDARs too, where layers are separated from each other, and the same object seen on the disjoint layers cannot be connected. However, sequentially registering information on each beam results in a similar problem we deal with. The line segments of the scrappy object can be categorized in each layer as separated curve for each scan.

This paper addresses the problem, where sparse 3D clouds can be built from sequentially scanned data, without having full 3D cloud. We will show here that this data gathering through motion may contain enough information for semantic level analysis of the neighborhood of the vehicle. Our method is capable of recognizing 3D shapes without having to look on the whole shape, or having dense resolution point clouds for sufficient modeling the shape or parts. In this paper we extend our earlier work [9] by some modification of our method aiming increase in its speed (real-time run is achieved) and precision, and also present results on a real-life urban database.

Approaching towards a street object, our method accumulates the information to increase the probability of detecting a possible object. Our method improves the system by using low-level scattered data sources for semantic level interpretation during the motion.

A tilted LIDAR-based decision support system for autonomous vehicles has several elements:

- The data of the LIDAR sensor has to be registered in a global coordinate system by fusing it with IMU, GPS or other synchronized localization data stream [10].
- The resulted enriched cloud should be preprocessed for later work, noise [11] and ghost removal [12] have to be done.
- Object candidates have to be segmented in order to classify the environment. This is usually done by connected component analysis or specific object detection [13].
- Shape classification is the next stage, which is investigated in details in the paper. At this stage we assume that we have some preliminary prediction about the surrounding objects, and we can also predict the environmental scenery type (e.g.: urban, warehouse, etc.). Using this assumption we can assign prior probability to different classes and we can make Bayes decision (e.g.: animals are unlikely in urban environment). In our tests on urban data of the proof-of-concept evaluation, we collected frequent urban objects and we assigned the same prior probability to them.
- In the final stage, using the classification result, some decision rule has to be applied to control the autonomous vehicle [14].

A. The Contribution of the Paper

SoA methods cannot be applied for recognition from small parts without shape model or incremental recognition and without exact scale information. Looking at the palette of the solutions of the best practices: applying conventional 3D recognition methods like [3] is not favorable, because mesh generation or voxelization steps can be expensive for incomplete data and can also cause information loss. 2.5D methods [15] are not applicable, because they cannot deal

[2] https://www.sick.com

with hidden points produced by the viewpoint change during motion. Model based approaches [16] also has to be excluded, because we only know the partial object.

We propose a new solution addressing the following issues:

- Processing steps work directly on point clouds, avoiding the mesh generation and possible information loss;
- Keypoint search based on local radius in 3D, making it independent of the full size;
- Local feature based object description, which is not model based;
- New local graph based descriptors;
- Object prescription with bag of features.

The solution of the above issues offer the potential of recognition from partial clouds and solving the problem of sequential data gaining of tilted single-layer/multi-layer LIDAR sensors. Some aspects of these issues we have addressed in [9], where preliminary results were introduced on the validation level. In the following we will show how discriminative local patterns can be used for the classification of partially visible objects.

B. The Outline of the Paper

Section II introduces related works. In Section III, we describe our proposed solutions in detail. To validate the proposed method, in Section IV and V, we will show results on known 3D data and on real life MLS point clouds. Finally in Section VI we will draw some conclusions.

II. Related Works

3D object recognition and classification has a broad research interest in several diverse fields of sciences (like medical sciences [17] or augmented reality [18]), and especially in mobile robotics, transportation and vehicle sciences [19]. For autonomous driving or surveillance, object recognition is an indispensable task, which makes conventional transportation systems to be smart. Before further analysis of sensors and methods we must differentiate the following range image and point cloud categories regarding in the following:

- full 3D: full 3D object surface is known (mainly computer mesh models).
- 3D: some hidden points are known, but full 3D object surface is unknown (generally acquired from registered frames, e.g. limited multiview series).
- 2.5D: we know only part of the object surface, which is visible from one viewpoint (point cloud generated from one frame, with depth sensors measuring x,y,z coordinates as well (kinect, 3D LIDAR)).
- 2D: planar contour points (measured by 2D LIDAR)
- partial cloud: with this we can refer to registered point cloud set (both from 2D or 2.5D), so generally it is in 3D, but contains less information than 2.5D clouds.

In the next subsections related works are surveyed regarding sensors, data structure and object recognition.

A. Sensors and Data Structure

Following the above categories, we overview the most common sensors and systems regarding point-cloud based object and environment detection methods. To reconstruct surroundings in 3D many general purpose depth sensors (kinect, ToF camera, stereo camera-pair) or methods which provide 3D information from 2D sensors (SfM) can be used. Some vehicles are equipped with LIDAR sensors, having its features of 360 (or near 360) degree angle of view and insensibility to lighting conditions [20].

1) 3D LIDARs: Using 3D or 2.5D scanning a wide range of application has been addressed in the recent years for autonomous robot navigation. These methods, including the sensors having 64 parallel beams (Velodyne HDL-64),[3] can achieve excellent results being standard for the possible best solutions. The application of these 3D (multi-planar or multi-layer) LIDARs is common in very different intelligent transportation systems. For example [21] realized localization based on curb and road marking detector, [22] localized and extracted street light poles with high recall value and in [23] traffic monitoring is done via 3D LIDAR sequences. Multi-layer LIDARs generate 2.5D information instantly, so in most cases processing is done on point cloud sequences in real-time instead of registering them, but the number of layers (the vertical resolution) and the angle of the beam opening (vertical view) might be too low for proper scanning. On the other hand, in intelligent vehicles combining 2D (planar or single-layer) LIDARs with pose sensors is still a relatively cheap and accurate solution for 3D reconstruction [24], [25]. Nowadays complete MLS systems are available for mapping purposes. They often fuse more than one LIDAR sensors and multiple cameras. Registering point clouds for mapping purpose is usually offline. There are already a few of them which can reconstruct the environment in real-time [26]. Stanley (DARPA Grand Challenge winner) [27] also used (multiple) single-layer LIDARs for building 3D environment in 2005. In case of AGVs multi-planar LIDARs have the disadvantage compared to planar ones that installing them needs additional cost, while 2D LIDARs are already operating on board as safety sensors.

2) 2D LIDARs: In research concerning transportation and mobile robotics, it is also common to apply a 2D LIDAR sensor for different purposes, like Simultaneous Localization and Mapping (SLAM) [28], detection and tracking [29]. 2D laser scanners are rarely applied standalone for these tasks, naturally 2.5D or 3D reconstruction is not even possible relying only on 2D data. Although there are some successful attempts: for example in [30], where pedestrian detection is implemented based on spatial-temporal walking patterns. Another example is [29], where humans were detected and tracked in mobile 2D LIDAR data. This was done by euclidean clustering of data points, cluster matching and making a heuristic based decision.

In general, these sensors exist as one component of diverse sensor networks or at least they are coupled with one additional sensor. Common realization of navigation related tasks is that the 2D laser scanner is complemented with sensors applicable for localization like GPS, INS, odometer, etc. This type of solutions allows transforming measured data to global coordinate system. In a mobile robot system capable of route

[3] http://velodynelidar.com/hdl-64e.html

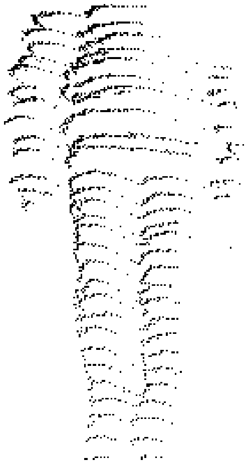


Fig. 3. Registered 2D LIDAR sequences about a pedestrian; in our scenario we detect only a limited number of 2D layers, practically started from the bottom scanning.

learning built by [31], the localization can be based on the signal of Wi-Fi access points and magnetic compass too.

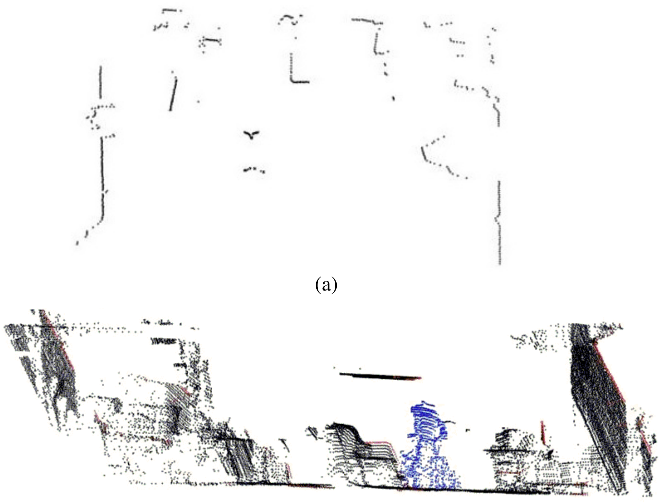
This category of 2D LIDARs are mostly for AGV safety systems and heavy trucks. In this case we can collect scanning data for a limited vertical viewing-angle at a given time, and the continuous collection of the registered data makes it possible to achieve some limited recognition information. This paper is about how to achieve additional recognition data from these limited viewing-angle scanning systems.

3) Sensor Fusion: Solving these tasks by sensor fusion is also a research direction. Shi *et al.* [32] propose a solution for pose estimation in urban areas where GPS data is inadequate. They realize the solution by using the fusion of 2D laser scanner and panoramic camera applying scale and loop closure constrained bundle adjustment. The fusion of a 2D laser scanner and an Asus Xtion depth sensor is applied as well. Trevor *et al.* [28] proposed a SLAM algorithm, which uses the detected planar surfaces as landmarks.

In the following we make the assumption that all the point clouds are made by one tilted 2D LIDAR or slices of 3D LIDAR (Fig. 3 and Fig. 4). The vertical information content of the latter case about the close angle environment is similar to the former one in the case of one frame. By this way we get sequential (bottom-up) exploration of an object.

B. Object Recognition in Point Clouds

Full 3D, 3D and 2.5D general object recognition pipelines can rely on either local or global descriptors. Local descriptors characterize well a given surface patch around a keypoint. Finding correspondences between these local surfaces can be the basis of object recognition. This can be done through different hypothesis generation and verification methods, but it is an exhausting search. A review about the state of the art of these pipelines is available in [38]. Contrary to this, global descriptors represent the whole object, so they are useful for object and category recognition in full 3D matching. There are global descriptors for 2.5D scanning as well [36]. Also local descriptors can be extended to global ones by simply considering the whole object cluster as a local neighborhood of a point [39]. One of the state of the art local descriptors is Rotational Projection Statistics (RoPS) [40]. RoPs outperforms former descriptors in feature matching (like [15]). However this method cannot be directly applied on point clouds: a mesh generation step is needed.



(a)

(b)

Fig. 4. Example of accumulated scanning during the AGV motion, building up a 3D point cloud from the plane LIDAR scans (Sensor position: altitude - 2 m, angle closed with the horizontal axis - 30°). (a) Actual scan in sensor coordinate system (SICK S300 Expert). (b) Registered scans in gloobal coordinate system; points of actual scans are indicated with red; points of recognized cyclist object indicated with blue.

Multi-planar LIDAR generated point clouds are considered as 2.5D as well, but these clouds are specific in terms of point density. In our case, 3D data generated by vehicle movement and information gain is relatively slow (but sequential). Regarding one frame, wide FoV multi-layer LIDARs can see the whole object. Tilted narrow FoV or 2D ones see much less. Solutions for whole objects [16] have no or only indirect applicability, in our problem objects are only partially present. In case of [37] the authors use GMM (Gaussian Mixture Model) to describe an object. In [41] features like geometrical shape and size, barycenter is used, also color of the object is commonly used, which we do not intend to use. The extraction of features what requires at least full ‘one-view’ of the object, cannot be extracted from partial clouds.

The main problem of classification based on single-layer 2D point clouds is the lack of surface information, thus lacking of distinctive properties. One way of solving this issue is the extension of the sensor system. For example in [19] visual features were also utilized for the categorization. These features can help in partial classification task, however they depend on visibility conditions. Lee *et al.* [42] tried to overcome on the problem of insufficient information by using multiple features like width of the object, range data variation and signal strength. This method proved to be efficient, but obviously from 3D data (which is achievable by registering the 2D clouds) more distinctive features are available. In [37] point classes (horizontal, vertical, slope, scatter) were defined

TABLE I
3D AND 2.5D RECOGNITION METHODS

Method	Example	Main application	Most common range image type	Applicability to incremental recognition from 2D laser scanners
Full 3D global shape recognition	STT [33], reeb graph [34]	classification, pose estimation	mesh	not applicable (there is no full 3D in real scenarios)
(Imperfect) 3D and 'globalized' local descriptors	2D descriptor based [35], GSH [36]	object recognition	point cloud, mesh	applicable (limited by information amount, rather suits to specific object than category recognition)
2.5D and 3D methods specialized for LIDAR point clouds	[16], [37]	detection, tracking	point cloud	not applicable (they are model based, rely on size of objects, which is unknown in early stages)
Semi-global descriptor (local pattern)	BoG (our proposal here)	object and category recognition	point cloud	applicable

based on consecutive point information and object classes were recognized based on these classes. They utilize GPS, IMU and wheel odometry together with three single-layer LIDARs to get 3D point clouds instead of 2D ones, as we propose too. Our proposed method define local patterns instead of points, so it is not so limited considering object category possibilities. An overview of shape recognition methods can be found in Table I. In this table, the last row contains our proposed method based on local pattern recognition, as explained in the following section.

III. THE PROPOSED METHOD

The proposed method compares statistics of local structures. The steps of the definition of patterns and matching are as follows:

- First, local surface is defined around each point. Saliency of the point is calculated by Harris operator on this surface.
- Then a local scale is assigned to significant points, it determines the number of keypoints. Different keypoint types will create local structures, so keypoints characterized by local surface descriptor are clustered.
- Local patterns are defined as graphs of keypoints.
- Finally, frequency of local patterns are compared.

We define Bag of Graphs (BoG) method, which is a kind of Bag of Features (BoF) [43]. In [9] we have examined different 3D descriptors; finally, we have chosen Bag of Graphs (BoG) as the most characteristic local descriptor, giving additional features and connectivity information. BoF methods are also used for probable pose/position estimation, as the evaluated features vote for the most probable characterizing position. We consider here rotational invariant object classification of partly nonrigid shapes, not considering the pose of the object.

The steps of BoG method are detailed below:

1) Preprocessing: There are required and optional preprocessing steps. The former ones are necessary to get object candidates from LIDAR sequences (registration, segmentation). Optional steps can be outlier removal or downsampling. For large point numbers we suggest to downsample the keypoint candidates from all the points of the cloud to the points marked as keypoints by Intrinsic Shape Signature (ISS) method [44].

2) Local Surface Definition: Based on an appropriate search radius, the corresponding neighborhood of a point represents the local surface around the point. This search radius must be chosen so that it will determine the smallest features we detect. Considering that 2D Harris detector is an effective method for salliency detection in 2D [45], extending it to 3D [46] a second order parametric surface is also fitted to these points. Here, Harris operator is applied and also used for further calculations (e.g. curvature estimation). The 3D version Harris operator defines z coordinate (instead of intensity like in 2D case) as a function of x and y.

$$f(x, y) = \frac{p_1}{2}x^2 + p_2xy + \frac{p_3}{2}y^2 + p_4x + p_5y + p_6, \quad (1)$$

where p_i $i = 1 \dots 6$ are parameters of the fitted surface.

The Harris matrix is [46]:

$$H = \begin{pmatrix} A & C \\ C & B \end{pmatrix}, \quad (2)$$

where

$$\begin{aligned} A &= p_4^2 + 2p_1^2 + 2p_2^2, &(3)\\ B &= p_5^2 + 2p_2^2 + 2p_3^2, &(4)\\ C &= p_4p_5 + 2p_1p_2 + 2p_2p_3. &(5)\end{aligned}$$

Before the surface fitting, Principal Component Analysis (PCA) is used for normal vector estimation [47]. Height function is aligned with the z-axis by rotating the points of local neighborhood and also the center is translated to the origin [46]. In that instance we search for parameters, p_i as the second-order contact approximation or 2-jet (truncated Taylor expansion). Higher order terms are ignored (they are 0 at the origin), p_4, p_5 will be zeros because of the translation and p_6 because of the alignment. For simplicity, we suppose the specific case where x and y were aligned to the principal directions as well, our coordinate frame would be the Monge coordinate system. This would result p_2 also to be 0 and makes p_1 and p_3 to be equal to the principal curvatures (k_1 and k_2) [48]. Examining this specific case, the relation between the eigenvalues of the Harris matrix (Harris curvatures) and the principal curvatures at the origin clearly

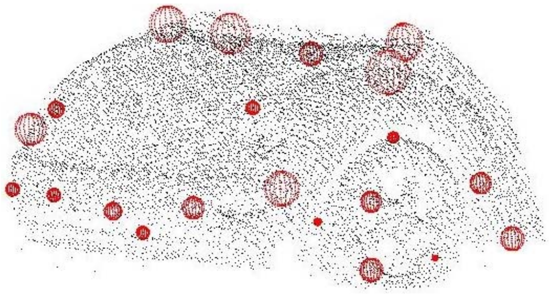


Fig. 5. Illustration of local scale in case of a car's hood and front wheel; we measure here the local Harris curvature around keypoints as local scale information.

can be seen:

$$H = \begin{pmatrix} 2p_1^2 & 0 \\ 0 & 2p_3^2 \end{pmatrix}. \tag{6}$$

$$k_i = \sqrt{\frac{\lambda_i}{2}}, \tag{7}$$

where λ_i-s are the eigenvalues of Harris matrix, and k_i-s are principal curvatures.

Note that, if the last assumption (x and y is aligned to the principal directions) is not valid (as in our general case), the Eq. (7) still can be deduced and holds.

3) Keypoint Search and Characteristic Radius: Salient, corner like points correspond to large Harris eigenvalues. For these points a local radius is defined. Eq. (7) provides a good basis to define our local radius in general case based on the definition of radius of curvature in specific case. By substituting the formula $\rho_i = \frac{1}{k_i}$ we get the relation of Harris curvatures and radius of curvatures. Latter is inversely proportional to the square root of the Harris curvatures:

$$\rho_1 = \sqrt{\frac{2}{\lambda_1}}, \tag{8}$$

where ρ_1 is the characteristic radius, and λ_1 is the smaller eigenvalue of the Harris matrix.

Selecting the final keypoints representing the local shape is based on this local radius. A new keypoint (found in descending salience order) can only be outside of characteristic spheres (environment) of former keypoints. These spheres are defined at previously found points by corresponding characteristic radii. Illustration of radii corresponding to different fineness structures is shown in Fig. 5.

4) Local Descriptor: Keypoints found in previous step can be characterized by local surface descriptors. Before calculating descriptors, normal vectors should be properly oriented by normal orientation propagation [49]. Here viewpoint based orientation is not applicable, because the viewpoint is continuously changing. The descriptor used by us is assembled from the following components:

- **Volume of local convex hull:** Scale information

$$V_c = \sum V_{t_i}, \tag{9}$$

where V_c is the volume of local convex hull. V_{t_i} is the volume of i-th tetrahedron corresponds to i-th triangle, building up the convex hull surface and its barycenter-origin distance. Note that, we use triangulation only locally, and just in the neighborhood of keypoints.

- **Characteristic radius:** Scale information

$$\rho_2 = \sqrt{\frac{2}{\lambda_2}}, \tag{10}$$

where ρ_2 is the characteristic radius, and λ_2 is the larger eigenvalue of the Harris matrix. ρ_1 was used for the identification of neighboring keypoints, but ρ_2 values also can be distinctive.

- **Surface normal angle:** Information about the effect of local scale change

$$cos(\theta_s) = \frac{n_{small} \cdot n_{large}}{\|n_{small\|} \, \|n_{large}\|}, \tag{11}$$

where θ_s is the surface normal angle, n_{small} and n_{large} are the normal vectors calculated with a smaller and larger neighborhood [50].

- **Modified shape index:** Local reference frame (LRF) invariant curvature proportion

$$I_{mod} = \left| \frac{1}{\pi} \arctan \frac{k_1 + k_2}{k_1 - k_2} \right|, \tag{12}$$

where I_{mod} is the modified shape index [51]. Elementary surface shapes are corresponds to values 0.5 (cup or cap), 0.25 (rut or ridge) and 0 (saddle).

- **Point Feature Histogram:** Density invariant generalization of the mean curvature, as a normalized histogram over the values all the point-pair measures α, ϕ and θ in the given neighborhood are defined as:

$$\begin{aligned} u_p &= n_s, \\ v_p &= u_p \times \frac{p_t - p_s}{\|p_t - p_s\|}, \\ w &= u_p \times v_p, \qquad (13) \\ \alpha &= w \cdot n_t, \\ \phi &= u_p \cdot \frac{p_t - p_s}{\|p_t - p_s\|}, \\ \theta &= \arctan\left(w \cdot n_t, u_p \cdot n_t\right), \qquad (14) \end{aligned}$$

where u_p, v_p and w vectors construct a local reference frame. p_t, p_s, n_t and n_s correspond to target and source points and their normal vectors [47]. It is worth mentioning that in order to speed-up the algorithm we used Fast Point Feature Histogram (FPFH) [52] instead of PFH in case of real-life database. The complexity of PFH is $O(nk^2)$ with n points and k number of neighbors, while the complexity of FPFH is $O(nk)$.

5) Keypoint Cluster Extraction: Clustering of keypoints is necessary to construct local patterns. In order to find keypoint clusters K-means is applied on local descriptor data corresponding to the keypoint database [53]. The cardinality of clusters has to be determined specifically to data sets based on the training results. It is a trade-off between homogeneity and separation.

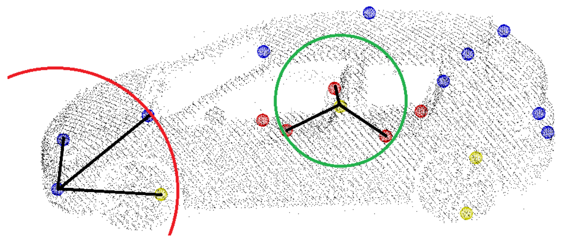


Fig. 6. Local pattern definition: the graph of clustered local keypoints is characteristic for the local structure.

6) Local Pattern Definition: Utilizing the keypoints, their clusters and Euclidean distance between the pairs of them, we can build a weighted non-homogeneous graph, which can represent a shape.

This undirected node-labeled graph by definition is a 4-tuple $g = (V, E, \sum, l)$, where V is the set of vertices (keypoints), $E \subseteq V \times V$ is the set of edges, $\sum$ is the alphabet of labels and $l : V \rightarrow \sum$ is a function mapping each vertex onto a label (this label generated by K-means in the previous step) [54].

Instead of using the graph of all the keypoints as a global descriptor, g_i is used as local pattern. We define $g_i = (V_i, E_i, \sum_i, l_i)$ subgraphs around each keypoint, where for the i-th vertex, $\upsilon \in V$, $\nu_n(\upsilon)$ denoting n-th nearest neighbor, $V_i = \{\nu_0(\upsilon), \nu_1(\upsilon), \ldots, \nu_{k-1}(\upsilon)\}$, using the definition of k nearest vertex to υ from [55]. These subgraphs are illustrated in Fig. 6. Four points were chosen to build g_i, so volume can be assigned as one more feature of the subgraph. We defined subgraph similarity based on the center point type, the volume category and counting the number of surrounding points from all types.

For example (Fig. 6) suppose that we have three clusters of keypoints (red - 1, blue - 2, yellow - 3), two volume types (smaller - subgraph indicated with green circle, larger - subgraph marked with red arc). The subgraph in the red arc can be represented by: {1 112 2}. In this code 1: center type; 112: surrounding point types (sorted); 2: volume category of the graph.

7) Bag of Graphs: Counting the frequency of the similar local patterns defined in the previous step we get a BoF [43] like descriptor. Based on the example of the previous subsection, we have 3*10*2=60 possible variations of patterns. Unfortunately, this can already result in a sparse histogram, and the number variation possibilities increase radically by raising cluster, volume categories. For solving this problem, only patterns occurring in the train set will be counted. Methods like dimension reduction or generating less graph types by hash function can be used as well. Another option for brief BoG representation is the concatenation of graph type histogram with graph volume histogram. For 4-5 object categories, the example illustrates well the number of pattern variations (several hundreds) we used, for large object category numbers keypoint cluster cardinality should be increased. Absolute (L1) distance is used for error measurement and nearest neighbor classification is done, using only the training cluster centers.

TABLE II

Categorization Results (in %) on the Full One-View Test Clouds (Overall Result: 83 %, Comparable to GSH [36])

Query / Result	Human	Chair	Table	Angular object	Quadruped
Human	**67**	17	11	0	5
Chair	0	**94**	6	0	0
Table	0	28	**67**	5	0
Angular object	0	6	0	**94**	0
Quadruped	6	0	0	0	**94**

TABLE III

Categorization Results (in %) on the Partial One-View Test Clouds; About 20 % of the Full 3D Object Is Visible, There Are No Methods to Compare With

Query / Result	Human	Chair	Table	Angular object	Quadruped
Human	**61**	6	6	11	16
Chair	5	**89**	6	0	0
Table	17	11	**50**	16	6
Angular object	0	17	0	**78**	5
Quadruped	44	0	6	0	**50**

IV. Validation on Known Public Reference Database

The usefulness of local pattern based descriptors has been demonstrated first in [9] on a relatively large database generated from [56]. This is done after selecting categories closest to application area (human, chair, table, angular - box like - objects, animals) and applying Hidden Point Removal (HPR) operator [57] on them. HPR is a standard way to generate 2.5D views from 3D point clouds, [58] also used it to test descriptor specific keypoint detector repeatability in different views.

Confusion matrix is counted for the recognition of these samples from different view-points (Table II), it can be compared to other published results [36]. Note that other methods do not deal with partial clouds similar to Table III, thus these cannot be compared.

The 2.5D test database contained more than 1000 samples from 90 different one-view clouds. We simulated the exploration in several stages both in bottom-up and top-down sequence. For example Table III shows the confusion matrix for (25 keypoint stage, where only about 20% of the full 3D cloud is visible). Fig. 7 illustrates an example for point clouds tested in Table II and III.

The average recognition result for full 'one-view' clouds is about 83%. It is comparable to results of Global Structure Histogram (GSH) (about 80% efficiency in similar circumstances) method [36], which represents an object as distribution of paths along the surface. For clouds containing about 20% of the full 3D object (less than half of the 'one-view' clouds) our method performed 66%. It is comparable to its full 'one-view' recognition performance (83%). High certainty object category prediction is achieved by our method regarding five object categories for such partial clouds other methods do not even deal with. After validating on public database,

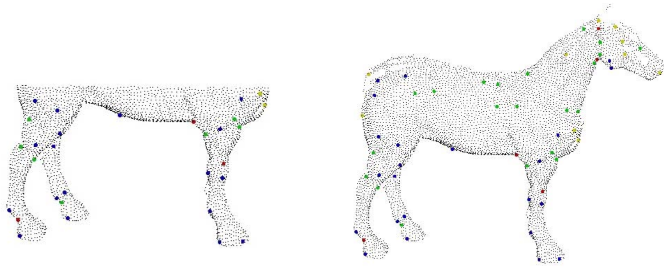


Fig. 7. Examples of test clouds generated from public database with clustered keypoints; from one view we can never see all the keypoints of a body, and moreover without scanning the full height we do not have scale information about the shape. (a) 20% of full 3D object. (b) 45% of full 3D object.

we step forward for testing the applicability in real life scenarios.

V. Real-Life Experiments

After the validation on a test dataset, real-life LIDAR sequences were chosen to generate the proof of concept of our method on noisy data. First the steps of test data generation, then the results will be presented.

A. Data Extraction

We built an object database containing segmented objects from Mobile Laser Scanning point cloud. In case of planar LIDAR sequences we get 3D clouds by registering them. This is done by using the position information of the vehicle. Registration of frames is inevitable in the case of 3D LIDAR frames too, because their sparsity and to get sequential structure. If position data is not available algorithms like Generalized-ICP can be used for registration [59]. This paper does not deal with segmentation process, known methods based on voxelization [60], or the method like [61] which uses adaptive radius for RBNN (radially bounded nearest neighbor) can be applied. Ground segmentation is generally the first step of the object segmentation algorithms based on voxel clustering by features like mean, variance and density. When the ground is segmented it operates as a separator between partitions determined by local neighborhood. One contribution of our approach is the ability to help this segmentation, because of prediction from partial clouds. Most of the literature seek to classify the urban object categories: {Road, Building, Tree, Car, Pole, Pedestrian, Cyclist} [37], [41]. Out of this seven categories only five: {Tree, Car, Pole, Pedestrian, Cyclist} have significance; ground is already found in the segmentation step and there are also solutions for automatic extracting of vertical building walls [62]. So we do not deal with points corresponding to Road and Building categories. Our MLS cloud was lack of cyclists, so we segmented cyclist clouds from the KITTI database [63], [64]. Cyclist category includes both bicyclists and motorcyclists. We trained our algorithm for a set of 70 3D objects (illustrated in Fig. 8) and tested it on 60 different objects, exploring them in bottom-up sequence in 5 steps. So, the test database consisted of total 300 clouds. In the test we explored objects only up to 2 m. In practice, it is likely we will only see object points above 2 m (depending on the sensor installation), if we pass the object. For objects lower than 2 m the 5 stage resulted stepsize of 20 % of object height, for objects taller than 2 m (Trees and Poles) the stepsize was exactly 0.4m. Tall objects were also explored up to full height in another test where we used ISS method to reduce the number of keypoint candidates and doubled search radius. This was necessary to deal with the increased number of points.

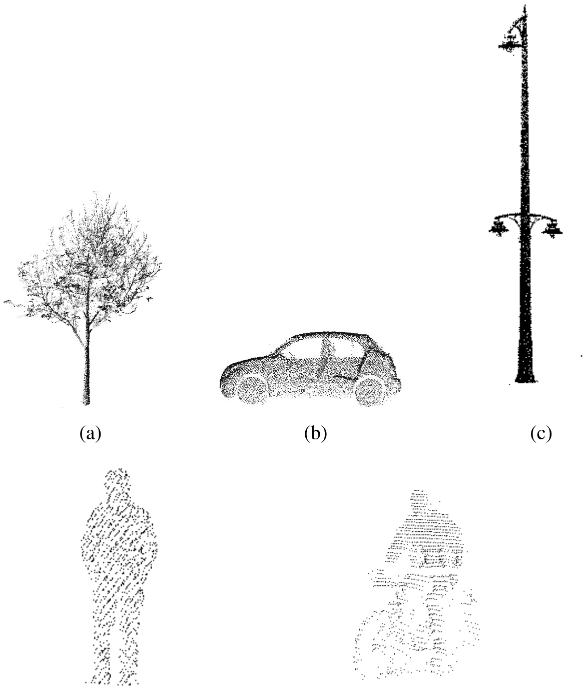


Fig. 8. Point cloud examples of trained objects. (a) Tree. (b) Car. (c) Pole. (d) Pedestrian. (e) Cyclist.

B. Computational Complexity and Running Speed Evaluation

We implemented our algorithm in C++ programming language and we used the running time of this code to evaluate the computational complexity of the algorithm. The ISS and FPFH functions are using PCL implementations [65].

Examining one-view clouds up to 2 m height, the average point number is about 6000. On an on-board computer with configuration: Intel Core i7-4790K @ 4.00GHz processor, 32 GB RAM, Ubuntu-16 64 bit operating system, the average running times of the main steps of our algorithm corresponding to the average point number is 0.354 s. Table IV shows detailed evaluation of a pipeline based on our method. The evaluated screen contains 60 consecutive measurement with a 2D LIDAR sensor (with 10 Hz scanning frequency) and the number of object points in the screen are the average indicated above.

Remarks:

- Running speed depends on the number of points, and the MLS point clouds we used for the evaluation are locally dense.
- A full one-view pedestrian with few hundred points can be processed and categorized with our algorithm

TABLE IV
RUNNING TIME EVALUATION ON AVERAGE POINT NUMBER

Program segment	Running time [ms]
Ground detection with RANSAC plane fit [65]	40
Object segmentation with euclidean clustering [65]	34
Our algorithm	280

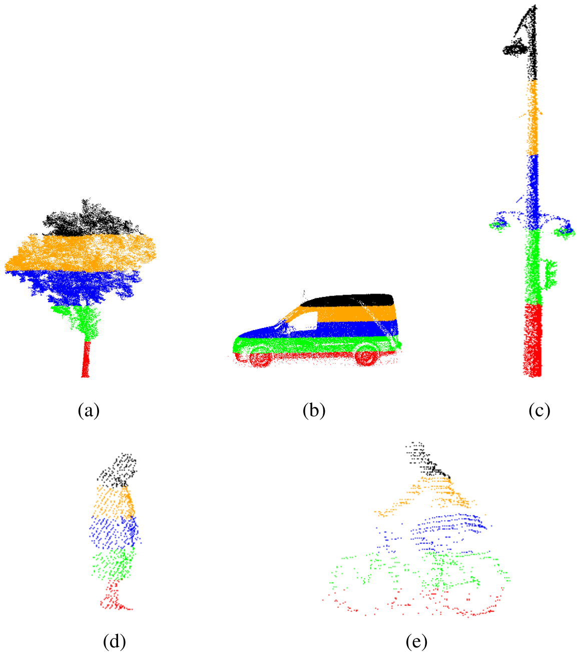


Fig. 9. Example point clouds of test object explorations; the point colors (red, green, blue, orange and black) correspond to newly registered points in the given exploration stage (20 %, 40 %, 60 %, 80 % and 100 % is visible of the object heigt) in case of each category. (a) Tree. (b) Car. (c) Pole. (d) Pedestrian. (e) Cyclist.

about 0.05 s with the current hardware and software configuration.

- Partial clouds have much less points than the calculated average, even in case of MLS data (however, it is also true, examining a tree up to full height, it can be built up from lot more points than its bole). Approximately 3 Hz (Table IV) is the lower bound for the process speed corresponds to full one-view clouds, for partial clouds it is faster.
- All the processing steps designed to be a part of parallel pipeline, running on multi-core (Registering frames, ground detection and object segmentation).
- AGVs are operating with a speed of walking, namely 1-1.5 $\frac{m}{s}$. Real-time and safe operation is ensured for them with our method by using a usual on-board computer.
- Implementing the code on computer with higher processing capacity (e.g.: NVIDIA DRIVE PX 2[4]computer designed for autonomous driving) would reduce the running speed at least one order of magnitude.

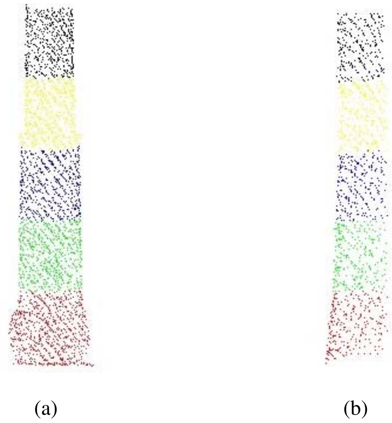


Fig. 10. Example point clouds of tree and pole test objects examining up to about 2 m height; the point colors (red, green, blue, yellow and black) correspond to newly registered points in the given exploration stage (0.4 m, 0.8 m, 1.2 m 1.6 m and 2 m is visible of the object height) in case of both category.

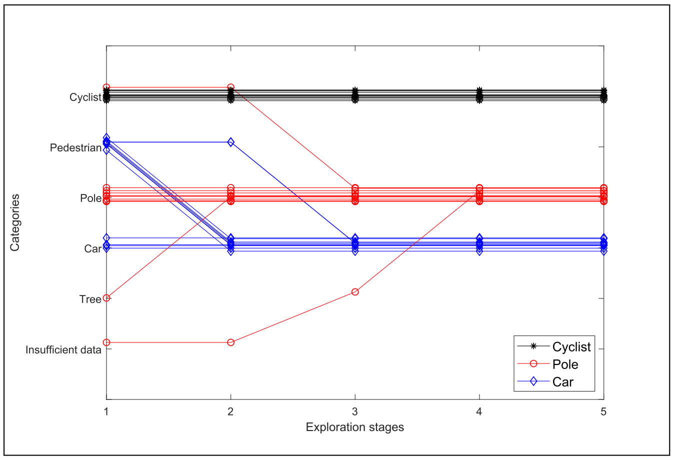


Fig. 11. Evolution of detection as the convergence of category changes through close-up stages for test objects of car, pole and cyclist categories. For objects lower than 2 m exploration stages are 20 %, 40 %, 60 %, 80 % and 100 % of object height, for objects taller than 2 m stages are 0.4 m 0.8 m 1.2 m 1.6 m 2 m.

TABLE V
RECOGNITION RESULTS IN DIFFERENT VIEW ANGLE STAGES (BOTTOM-UP SEQUENCES) ON URBAN OBJECTS FOR OBJECTS LOWER THAN 2 m EXPLORATION STAGES ARE 20 %, 40 %, 60 %, 80 % AND 100 % OF OBJECT HEIGHT, FOR OBJECTS TALLER THAN 2 m STAGES ARE 0.4 m 0.8 m 1.2 m 1.6 m 2 m

Exploration stage	1	2	3	4	5
Result (in %)	53.3	73.3	80.0	80.0	88.3

C. Results on Developing Point Clouds

Illustration of exploration of partial test point clouds are visible in Fig. 9. Overall recognition results for all stages are summarized in Table V. Decisions in each stages can be seen for the two least succesful categories in Fig. 12 (tree) and Fig. 13 (pedestrian) and for the other three in Fig. 11. In case of pedestrian category the recognition rate in the final stage is 91.67 % as well. Pedestrian detection works robustly in 2D,

[4] http://www.nvidia.com/object/drive-px.html

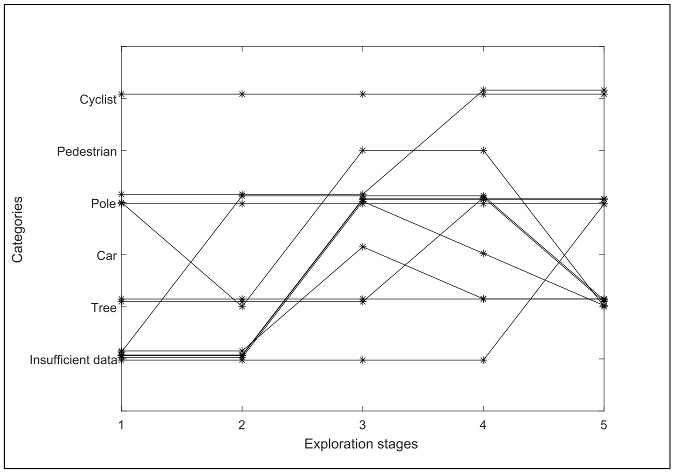


Fig. 12. Evolution of detection as the convergence of category changes through close-up stages up 2 m height for test objects of tree category. For objects lower than 2 m exploration stages are 20 %, 40 %, 60 %, 80 % and 100 % of object height, for objects taller than 2 m stages are 0.4 m 0.8 m 1.2 m 1.6 m 2 m.

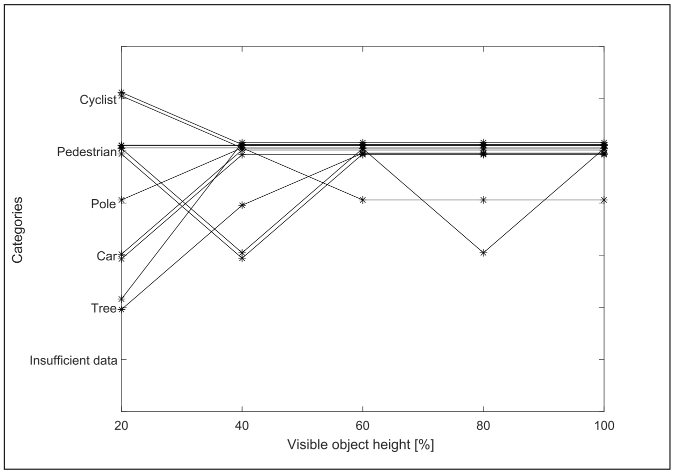


Fig. 13. Evolution of detection as the convergence of category changes through close-up stages for test objects of pedestrian categories.

if the occlusion is less than 30 %. [66], [67] In case of trees (without leaves) investigating parts under 2 m the category uncertain, they frequently categorized as poles (Fig. 13). Yet, it is just that what we would expect. Taking a look at the pictures of such tree and pole objects on Fig. 10, there is no apparent difference between these objects in the early stages. When the canopy appears, the method instantly differentiate the two categories. If we have data above 2 m we can continue the exploration up to full height. We did this by increasing search radius for keeping processing time. However, applying big radius for the whole exploration process is not advised, because it requires more data for evaluation.

If height information is available, it is advised to use small search radius and after reaching a given altitude (for example 2 m) increase it, and compare only categories that can be higher than the given altitude. In our experiments adding this binary decision variable **98.33 %** of the test clouds are categorized well when we reached full height of the objects.

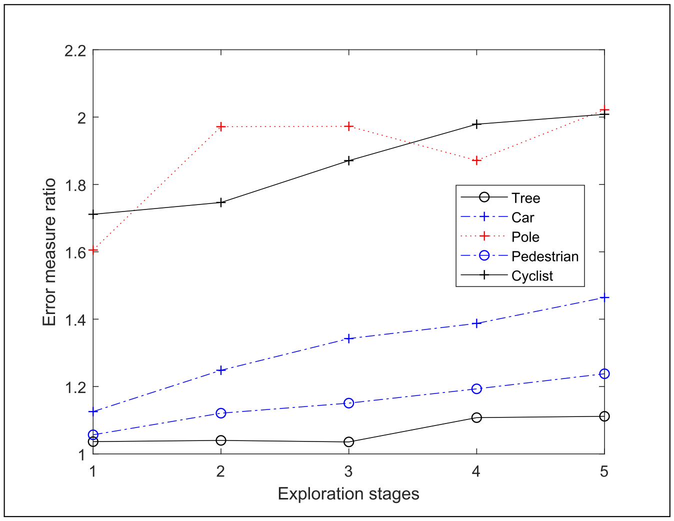


Fig. 14. Evolution of average error measure ratios for true positives. For objects lower than 2 m exploration stages are 20 %, 40 %, 60 %, 80 % and 100 % of object height, for objects taller than 2 m stages are 0.4 m 0.8 m 1.2 m 1.6 m 2 m.

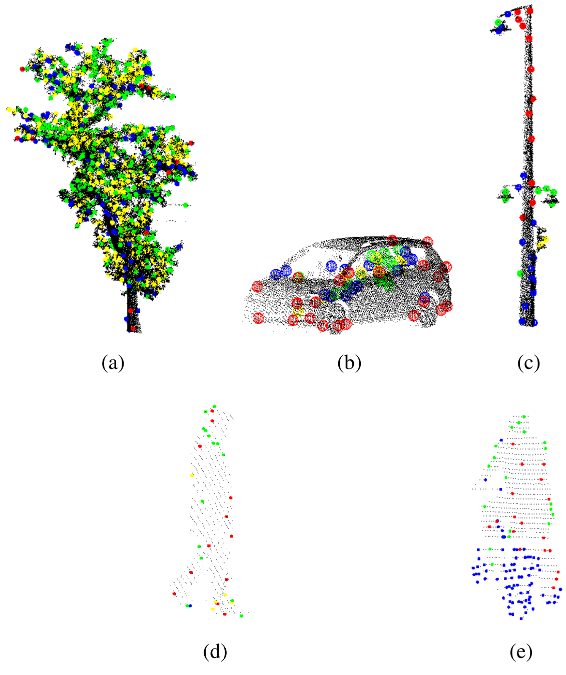


Fig. 15. Illustration of characteristic keypoint types. (a) Tree. (b) Car. (c) Pole. (d) Pedestrian. (e) Cyclist.

Characteristic keypoint types of different categories are illustrated in Fig. 15. One can observe that cyclists sitting on the bicycle are very similar to pedestrians, but cyclist category is still easily distinguishable from low information, because of the properties of bicycle.

Fig. 14 illustrates the certainties of the decisions with the evolution of average error measure ratios in true positive cases. This ratio is defined as:

$$\text{Error measure ratio} = \frac{\text{Second smallest error measure}}{\text{Error measure of true category}}.$$

For some categories we can make confident decisions in the first exploration stage, prediction of the correct object type happens in early stages. The results are even more promising than we got from the earlier public database. In this test we differentiated four keypoint clusters and four graph volume types, the final descriptor was a normalized histogram with 84 bins.

We investigated convergence, convergence speed and stability of the method, and we get satisfying results exemplified in the figures of this chapter. Summarizing, correct and confident decision can be made about obstacles based on real, partial point clouds.

VI. Conclusion

In the paper we proposed a 3D recognition method exploiting local information of point clouds. This method can solve the problem of partial view and partial shape detection problem for using the scanned point clouds of autonomous vehicles. We demonstrated that our method is capable of dealing with both synthetic and real partial point clouds. This method has the advantages compared to other methods applied on LIDAR point clouds earlier:

- It works directly on point clouds: Extra processing steps and information loss of local structures can be avoided.
- It is not model based: While model based methods are strongly restricted, when it comes to objects which radically differ from the model, our method is based on feature learning and local pattern recognition.
- It is rotation invariant: With rotation invariance unusual object position can also be detected like fallen trees or overturned cars.
- It can recognize objects from very partial clouds: It is the main contribution of our paper. As our best knowledge it is the first time to attempt the recognition of objects from such low information as we did. This allows us to integrate it with some segmentation method and mutually benefit from each other. Applying the method with a safety scanner - localization sensor system, early prediction can enhance security in large extent.
- Our method gives comparable results on one-view full size objects to other methods. Despite of the fact, we designed it to classify partial clouds (in order to predict category in early stage), not full one-view scans.

At present, in case of AGV-s our algorithm works in realtime on on-board computers. In urban environment, the operating speed of cars is one magnitude greater, so it requires special computers designed for autonomous driving for realtime running. Further possibility for speeding up our method can be the differential processing of point clouds. Dealing with only the actual scan and exploiting the evaluation results from previously registered frames can significantly reduce the necessary computing capacity. In the following we will examine our method in sophisticated traffic conditions of AGVs, where we know more about the a priori probabilities of occurrences of different objects. Beside that, we will investigate the applicability of the method on very sparse point clouds and also the decision stage of the BoG approach is planned to be supported with pattern sequence information.

Acknowledgment

The authors would like to thank the Budapest Road Management Department for providing Riegl VMX-450 Mobile Laser Scanning data sets.

The authors would also like to thank the Department of Material Handling and Logistics Systems of the Budapest University of Technology and Economics for providing SICK S300 Expert sensor and Creform AGV-A35046 for their measurements.

References

[1] A. Mukhtar, L. Xia, and T. B. Tang, “Vehicle detection techniques for collision avoidance systems: A review,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2318–2338, Oct. 2015.

[2] J. Yu, Z. Cai, and Z. Duan, “Detection of static and dynamic obstacles based on fuzzy data association with laser scanner,” in *Proc. 4th Int. Conf. Fuzzy Syst. Knowl. Discovery (FKSD)*, vol. 4. Aug. 2007, pp. 172–176.

[3] M. Pedersoli and T. Tuytelaars, “Learning where to position parts in 3D,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 136–144.

[4] P. B. Pascoal *et al.*, “Retrieval of objects captured with Kinect one camera,” in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2015, pp. 145–151.

[5] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view Convolutional Neural Networks for 3D Shape Recognition,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[6] R. J. López-Sastre, A. García-Fuertes, C. Redondo-Cabrera, F. Acevedo-Rodríguez, and S. Maldonado-Bascón, “Evaluating 3D spatial pyramids for classifying 3D shapes,” *Comput. Graph.*, vol. 37, no. 5, pp. 473–483, 2013.

[7] Z. Lian *et al.*, “Non-rigid 3D Shape Retrieval,” in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2015, pp. 107–120.

[8] M. Savva *et al.*, “Large-scale 3D shape retrieval from ShapeNet core55,” in *Proc. Eurograph. Workshop 3D Object Retrieval*, 2016, pp. 1–11.

[9] T. Sziranyi and Z. Rozsa, “Exploring in partial views: Prediction of 3D shapes from partial scans,” in *Proc. Int. Conf. Control Autom. (ICCA)*, Jun. 2016, pp. 707–713.

[10] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias, “Review of mobile mapping and surveying technologies,” *Measurement*, vol. 46, no. 7, pp. 2127–2145, 2013.

[11] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D point cloud based object maps for household environments,” *Robot. Auto. Syst.*, vol. 56, no. 11, pp. 927–941, 2008.

[12] B. Nagy and C. Benedek, “3D CNN based phantom object removing from mobile laser scanning data,” in *Proc. Int. Joint Conf. Neural Netw.*, May 2017, pp. 4429–4435.

[13] Y. Belkhouche, P. Duraisamy, and B. Buckles, “Graph-connected components for filtering urban LiDAR data,” *J. Appl. Remote Sens.*, vol. 9, no. 1, p. 096075, 2015.

[14] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, “MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1670–1677.

[15] X. Li and I. Guskov, “Multi-scale features for approximate alignment of point-based surfaces,” in *Proc. 3rd Eurograph. Symp. Geometry Process.*, 2005, pp. 217–226.

[16] A. Borcs, B. Nagy, M. Baticz, and C. Benedek, “A model-based approach for fast vehicle detection in continuously streamed urban LIDAR point clouds,” in *Computer Vision*. Cham, Switzerland: Springer, 2015, pp. 413–425.

[17] I. Atmosukarto, K. Wilamowska, C. Heike, and L. G. Shapiro, “3D object classification using salient point patterns with application to craniofacial research,” *Pattern Recognit.*, vol. 43, no. 4, pp. 1502–1517, 2010.

[18] W. Lee, N. Park, and W. Woo, “Depth-assisted real-time 3D object detection for augmented reality,” in *Proc. Int. Conf. Artif. Reality Telexistence*, 2011, pp. 126–132.

[19] H.-C. Moon, J.-H. Kim, and J.-H. Kim, “Obstacle detecting system for unmanned ground vehicle using laser scanner and vision,” in *Proc. Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2007, pp. 1758–1761.

[20] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2011, pp. 163–168.

[21] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer LIDAR," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 420–429, Feb. 2016.

[22] F. Wu *et al.*, "Rapid localization and extraction of street light poles in mobile LiDAR point clouds: A supervoxel-based approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 292–305, Feb. 2017.

[23] C. Benedek, "3D people surveillance on range data sequences of a rotating Lidar," *Pattern Recognit. Lett.*, vol. 50, pp. 149–158, Dec. 2014.

[24] J. R. Rosell *et al.*, "Obtaining the three-dimensional structure of tree orchards from remote 2D terrestrial LIDAR scanning," *Agricultural Forest Meteorol.*, vol. 149, no. 9, pp. 1505–1515, 2009.

[25] Á. Llamazares, E. J. Molinos, M. Ocaña, L. M. Bergasa, N. Hernández, and F. Herranz, *3D Map Building Using a 2D Laser Scanner*. Berlin, Germany: Springer, 2012, pp. 412–419.

[26] Y. K. Wang, J. Huo, and X. S. Wang, "A real-time robotic indoor 3D mapping system using duel 2D laser range finders," in *Proc. 33rd Chin. Control Conf. (CCC)*, Jul. 2014, pp. 8542–8546.

[27] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge," in *The DARPA Grand Challenge: The Great Robot Race*. Berlin, Germany: Springer, 2007, pp. 1–43.

[28] A. J. B. Trevor, J. G. Rogers, and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 3041–3048.

[29] T. Taipalus and J. Ahtiainen, "Human detection and tracking with knee-high mobile 2D LIDAR," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2011, pp. 1672–1677.

[30] X. Shao, H. Zhao, K. Nakamura, K. Katabira, R. Shibasaki, and Y. Nakagawa, "Detection and tracking of multiple pedestrians by using laser range scanners," in *Proc. IROS IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 2174–2179.

[31] V. Alvarez-Santos, A. Canedo-Rodriguez, R. Iglesias, X. M. Pardo, C. V. Regueiro, and M. Fernandez-Delgado, "Route learning and reproduction in a tour-guide robot," *Robot. Auto. Syst.*, vol. 63, pp. 206–213, Jan. 2015.

[32] Y. Shi *et al.*, "Fusion of a panoramic camera and 2D laser scanner data for constrained bundle adjustment in GPS-denied environments," *Image Vis. Comput.*, vol. 40, pp. 28–37, Aug. 2015.

[33] D. Zarpalas, P. Daras, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis, "3D model search and retrieval using the spherical trace transform," *EURASIP J. Adv. Signal Process.*, vol. 2007, p. 023912, Dec. 2006.

[34] T. Tung and F. Schmitt, "The augmented multiresolution reeb graph approach for content-based retrieval of 3D shapes," *Int. J. Shape Model.*, vol. 11, no. 1, p. 91, 2005.

[35] H. Dutagaci, A. Godil, B. Sankur, and Y. Yemez, "View subspaces for indexing and retrieval of 3D models," *Proc. SPIE*, vol. 7526, pp. 75260M-1–75260M-12, Feb. 2010.

[36] M. Madry, C. H. Ek, R. Detry, K. Hang, and D. Kragic, "Improving generalization for 3D object categorization with global structure histograms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 1379–1386.

[37] Y. Choe, S. Ahn, and M. J. Chung, "Online urban object recognition in point clouds using consecutive point information for urban robotic missions," *Robot. Auto. Syst.*, vol. 62, no. 8, pp. 1130–1152, 2014.

[38] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.

[39] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. 23rd IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2010, pp. 2155–2162.

[40] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, 2013.

[41] A. K. Aijazi, A. Serna, B. Marcotegui, P. Checchin, and L. Trassoudaine, *Segmentation and Classification of 3D Urban Point Clouds: Comparison and Combination of Two Approaches*. Cham, Switzerland: Springer, 2016, pp. 201–216.

[42] M. Lee, S. Hur, and Y. Park, "An obstacle classification method using multi-feature comparison based on 2D LIDAR database," in *Proc. 12th Int. Conf. Inf. Technol. New Generat. (ITNG)*, Apr. 2015, pp. 674–679.

[43] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. Workshop Statist. Learn. Comput. Vis. (ECCV)*, 2004, pp. 1–22.

[44] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV)*, Sep. 2009, pp. 689–696.

[45] A. Kovács and T. Szirányi, "Improved harris feature point set for orientation-sensitive urban-area detection in aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 796–800, Jul. 2013.

[46] I. Sipiran and B. Bustos, "Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes," *Vis. Comput.*, vol. 27, no. 11, pp. 963–976, Nov. 2011.

[47] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Dept. Comput. Sci., Tech. Univ. Munich, Munich, Germany, Oct. 2009.

[48] F. Cazals and M. Pouget, "Estimating differential quantities using polynomial fitting of osculating jets," *Comput. Aided Geometric Design*, vol. 22, no. 2, pp. 121–146, 2005.

[49] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, Jul. 1992.

[50] A. Flint, A. Dick, and A. V. D. Hengel, "Thrift: Local 3D structure recognition," in *Proc. 9th Biennial Conf. Austral. Pattern Recognit. Soc. Digit. Image Comput. Techn. Appl.*, Dec. 2007, pp. 182–188.

[51] T. Zaharia and F. Preteux, "Three-dimensional shape-based retrieval within the MPEG-7 framework," *Proc. SPIE*, vol. 4304, pp. 133–145, Jan. 2001.

[52] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, Piscataway, NJ, USA, May 2009, pp. 1848–1853.

[53] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 2006.

[54] R. Di Natale, A. Ferro, R. Giugno, M. Mongiovì, A. Pulvirenti, and D. Shasha, "SING: Subgraph search in non-homogeneous graphs," *BMC Bioinformatics*, vol. 11, no. 1, p. 96, 2010.

[55] M. Jiang, A. W.-C. Fu, and R. C.-W. Wong, "Exact top-k nearest keyword search in large networks," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, New York, NY, USA, 2015, pp. 393–404.

[56] R. C. Veltkamp and F. B. T. Haar, "SHREC2007: 3D shape retrieval contest," Dept. Inf. Comput. Sci., Utrecht Univ., Utrecht, The Netherlands, Tech. Rep. UU-CS-2007-015, 2007.

[57] S. Katz and A. Tal, "On the visibility of point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1350–1358.

[58] S. Salti, F. Tombari, R. Spezialetti, and L. D. Stefano, "Learning a descriptor-specific 3D keypoint detector," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2318–2326.

[59] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robot., Sci. Syst.*, vol. 2, no. 4, p. 435, Jun. 2009.

[60] B. Douillard *et al.*, "On the segmentation of 3D LIDAR point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 2798–2805.

[61] Y. Choe, S. Ahn, and M. J. Chung, "Fast point cloud segmentation for an intelligent vehicle using sweeping 2D laser scanners," in *Proc. 9th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Nov. 2012, pp. 38–43.

[62] M. Rutzinger, S. O. Elberink, S. Pu, and G. Vosselman, "Automatic extraction of vertical walls from mobile and airborne laser scanning data," *Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 38, no. 3, p. W8, 2009.

[63] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 3354–3361.

[64] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[65] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Shanghai, China, May 2011, pp. 1–4.

[66] D. Varga and T. Sziranyi, "Detecting pedestrians in surveillance videos based on convolutional neural network and motion," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2016, pp. 2161–2165.

[67] D. Varga and T. Sziranyi, "Robust real-time pedestrian detection in surveillance videos," *J. Ambient Intell. Humanized Comput.*, vol. 8, no. 1, pp. 79–85, Feb. 2017. [Online]. Available: https://doi.org/10.1007/s12652-016-0369-0

Zoltan Rozsa (M'18) received the M.Sc. degree in mechanical engineering modeling from Budapest University of Technology and Economics, where he is currently pursuing the Ph.D. degree with the Department of Material Handling and Logistic Systems. He is also a Research Assistant with the Machine Perception Research Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences. His research interests include construction logistics, material handling machines, machine vision, and 3-D recognition.

Tamas Sziranyi (SM'04) received the Ph.D. and D.Sci. degrees from the Hungarian Academy of Sciences, Budapest, in 1991 and 2001, respectively. He was a Full Professor with University of Pannonia, Veszprém, Hungary, in 2001 and Pázmány Péter Catholic University, Budapest, in 2004. He is currently a Full Professor with Budapest University of Technology and Economics. He has been a Research Scientist with the Institute for Computer Science and Control, Hungarian Academy of Sciences, since 1992, where he has been leading the Machine Perception Research Laboratory since 2006. With his Research Laboratory, he has participated in several prestigious international projects, including ESA, EDA, FP6, FP7, and OTKA. His research activities include machine perception, pattern recognition, texture and motion segmentation, Markov random fields and stochastic optimization, remote sensing, surveillance, intelligent networked sensor systems, graph-based clustering, and digital film restoration.

He has over 270 publications, including 50 in major scientific journals and several international patents. He has been fellow of the International Association of Pattern Recognition and the Hungarian Academy of Engineering since 2008. He was honored by the Master Professor Award in 2001, the Szechenyi Professorship, and the Pro Scientia (Veszprem) Award in 2011. He was the Founder of the Hungarian Image Processing and Pattern Recognition Society, where he was also the President from 1997 to 2002. He was an Associate Editor (AE) of IEEE TRANSACTIONS ON IMAGE PROCESSING from 2003 to 2009. He has been an AE of *Digital Signal Processing* (Elsevier) since 2012.