Extraction of Vehicle Groups in Airborne Lidar Point Clouds with Two-Level Point Processes

Attila Börcs, *Student Member, IEEE,* Csaba Benedek, *Member, IEEE,*

***Abstract*—In this paper we present a new object based hierarchical model for joint probabilistic extraction of vehicles and groups of corresponding vehicles – called *traffic segments* – in airborne Lidar point clouds collected from dense urban areas. Firstly, the 3-D point set is classified into terrain, vehicle, roof, vegetation and clutter classes. Then the points with the corresponding class labels and echo strength (i.e. intensity) values are projected to the ground. In the obtained 2-D class and intensity maps we approximate the top view projections of vehicles by rectangles. Since our tasks are simultaneously the extraction of the rectangle population which describes the position, size and orientation of the vehicles and grouping the vehicles into the traffic segments, we propose a hierarchical, Two-Level Marked Point Process (L^2MPP) model for the problem. The output vehicle and traffic segment configurations are extracted by an iterative stochastic optimization algorithm. We have tested the proposed method with real data of a discrete return Lidar sensor providing up to four range measurements for each laser pulse. Using manually annotated Ground Truth information on a data set containing 1009 vehicles, we provide quantitative evaluation results showing that the L^2MPP model surpasses two earlier grid-based approaches, a 3-D point-cloud-based process and a single layer MPP solution. The accuracy of the proposed method measured in F-rate is 97% at object level, 83% at pixel level and 95% at group level.**

***Index Terms*—Lidar, aerial laser scanning, vehicle, urban, Marked Point Process**

I. Introduction

Analyzing the vehicle populations of inner city areas is a central goal of automatic traffic monitoring and control, environmental protection and aerial surveillance applications [1]. To obtain a complex scene description, we need a hierarchical modeling approach. At low level *individual vehicles* should be detected and separated with accurate size and orientation estimation. At a higher level we need to identify the groups of corresponding vehicles, called hereafter *traffic segments*, such as cars in a parking lot, or a vehicle queue waiting in front of a traffic light. Corresponding automated approaches in the literature can be grouped first based on the used sensors and measurements; second based on the software modules focusing on the applied signal processing and artificial intelligence algorithms.

The authors are with the Distributed Events Analysis Research Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), H-1111 Kende u. 13-17 Budapest, Hungary. E-mail: {attila.borcs,csaba.benedek}@sztaki.mta.hu. A. Börcs is also with the Department of Control Engineering and Information Technology (IIT), Budapest University of Technology and Economics.

This work was partially funded by the Government of Hungary through a European Space Agency (ESA) Contract under the Plan for European Cooperating States (PECS), and by the Hungarian Research Fund (OTKA #101598). C. Benedek was also supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

A. Sensing technologies

Various sensing technologies have already been utilized for vehicle monitoring. Beside terrestrial sensors such as video cameras and induction loops, airborne and spaceborne data sources are frequently used to support the scene analysis. Dealing with optical imagery, recent vehicle detection methods exploit the improving quality and resolution of the obtained aerial or satellite images [2], [3], [4]. Long time thermal infrared (TIR) cameras are used for traffic monitoring due to their 'day-and-night' capability and their potential to derive temperature and temperature differences of objects [5], [6], [7]. Traffic surveillance is also an important civilian application of radars [8], which have the advantage of jointly providing the location and speed of the vehicles. Efficient radar based solutions have been proposed for monitoring non-urban roads or highways [9], [10] from remote sensing platforms, or city centers from terrestrial installations [8]. SAR images can also be used to detect stationary vehicles [11]. A comprehensive overview on the above mentioned aerial technologies for traffic estimation can be found in [5], [12].

The Light Detection and Ranging (Lidar) technology offers an efficient alternative solution for vehicle detection since it can jointly provide an accurate 3-D geometrical description of the scene, and additional features about the reflection properties and structures of the surfaces.

In this paper we deal with measurements of an aerial discrete return (DR) Lidar sensor [13], which is able to capture up to four range measurements for a single laser pulse, including 1st, 2nd, 3rd and last returns. We may also obtain four intensity returns of each pulse, which are related to the strength of the backscattered echoes. The intensity calibration step [14], performed by a commercial software, is considered as a black-box module by our processing methods. The density of the collected point clouds is around 8 points/m^2.

B. Related work on Lidar based vehicle detection

Lidar based vehicle detection methods in the literature follow generally either a grid-based or a 3-D point-cloud-based approach [15]. In the first group of techniques [16], [17], the obtained Lidar data is first transformed into a dense 2.5-D Digital Elevation Model (DEM). Thereafter various image processing operations can be adopted to extract the vehicles, such us thresholding [16], watershed segmentation [18] or morphology based connected component analysis [16]. On the other hand, in point cloud based methods [1], the feature extraction and recognition steps work directly on the 3-D point clouds. In this way we avoid loosing information due to projection and interpolation, however the time and memory

requirement of the processing algorithms may be significantly higher.

Another important factor is related to the types of measurements utilized in the detection. A couple of earlier works combined multiple data sources, e.g. [19] fused Lidar and electro-optical camera inputs. Other methods rely purely on geometric information [17], [18], emphasizing that these approaches do not depend on the accuracy of image-to-point-cloud registration. Regardless of the difficulties with radiometric calibration [14], the Lidar intensity is often used as an auxiliary channel in terrain classification and object detection tasks [20], [21]. Nevertheless, the intensity-related parameters of the classification process must be carefully set for specific Lidar devices, calibration techniques, and capturing circumstances.

While most of the Lidar based vehicle detection methods focus on static scenarios, in [1], [15] motion information has been extracted from the *shearing* distortion of the observed vehicle shapes. This approach exploits that due to the sequential line scanning technology applied in aerial Lidar scanners, moving vehicles from top view appear as parallelograms instead of rectangles in the point clouds. A binary shape classification method has been introduced in [22] to group the objects based on the estimated velocity. However it has also been noted that it is often difficult to decide whether the observed shape distortion is caused by target motion or missing data, yielding a number of detected objects with the status 'uncertain motion'. We have also experienced in various data sets that the relevance of this feature may depend on the data quality, the speed of the traffic flow, the sensor position w.r.t. the target motion and the scanning frequency of the laser beam.

The vehicle detection techniques should also be examined from the point of view of object recognition methodologies. Machine learning methods offer noticeable solutions, e.g. [17] adopts a cascade AdaBoost framework to train a classifier based on edgelet features. However, the authors also mention that it is often difficult to collect enough representative training samples, therefore, they generate more training examples by shifting and rotating a few training annotations. Model based methods attempt to fit 2-D or 3-D car models to the observed data [1], however, these approaches may face limitation for low resolution point clouds with complex and highly various vehicle shapes.

We can also group the existing object modeling techniques whether they follow a *bottom-up* or an *inverse* (i.e. a top-down) approach. The *bottom-up* techniques usually consist in extracting *primitives* (blobs, edges, corners etc.) and thereafter, the objects are constructed from the obtained features by a sequential process. To extract the vehicles, [16] introduce three different methods with similar performance results, which combine surface warping, Delaunay triangulation, thresholding and Connected Component Analysis (CCA). [18] apply the h-maxima transform followed by watershed segmentation to separate the objects. The output is a set of vehicle contours, however, some car silhouettes are only partially extracted and a couple of neighboring objects are merged into the same blob. In general, bottom-up techniques can be relatively fast, however construction of appropriate primitive filters may be difficult/inaccurate, and in the sequential workflow, the failure of each step may corrupt the whole process. In addition, we have limited options here to incorporate a priori information (e.g. shape, size) and object interaction.

Inverse methods [23] assign a fitness value to each possible object configuration, thereafter an optimization process attempts to find the configuration with the highest confidence. In this way complex object appearance models can be used, and it is easy to incorporate prior shape information (e.g. only searching among rectangles) and object interactions (e.g. penalizing intersection, favoring similar orientation). However, high computational need is present due to searching in the high dimensional population space. Therefore, applying efficient optimization techniques is a crucial need.

C. Involvement of road network information

The previously discussed techniques focus on extracting and analyzing individual vehicles, which can be achieved without considering complex structural models of the city layouts. However, for implementing a higher level traffic monitoring system, the utilization of the road network information becomes a necessary step, since the context of the vehicles can only be interpreted based on the neighborhood. The situation is simpler, if the scene consists of straight roads, so that an efficient traffic segmentation can be obtained by orientation based vehicle clustering [24]. This assumption has been exploited by us using data samples from Budapest, Hungary. However for a general usage of the model, we also need to provide strategies to deal with arbitrary road networks containing roundabouts and strongly curved roads.

There have recently been proposed a few approaches on complete road network extraction from airborne Lidar data [25], [26]. The Junction-Point Processes introduced in [27] may also give us a powerful tool for the problem with appropriate Lidar-specific modifications. Some of the existing techniques exploit the intensity channel of the Lidar measurements, assuming that asphalt provides usually lower intensities than vegetation [21], [25]. However as noted earlier the intensity calibration issue may mean a bottleneck here to use these methods for various types of sensors.

In this paper, we do not detail the road network extraction task, but we assume that either the scene contains only straight roads; or a coarse line network is available and registered to the Lidar data by using an automatically obtained or manually labeled city road map. This network will help us in the determination and analysis of possible interacting vehicles. However, this prior map solves neither the accurate terrain extraction nor the vehicle detection problems which should be still handled in an automated way.

D. Methodological contributions of the proposed approach

In our approach, we propose a hybrid model, where the initial point cloud is classified via 3-D features, but the optimal object configuration is extracted on a 2-D lattice, after ground plane projection.

Taking an energy minimization based approach we model traffic scenes by Marked Point Processes (MPP) [23], [28].

MPPs have previously been used for various population counting problems, dealing with a large number of objects which have low varieties in shape. Among alternative techniques with similar goals, we can mention Hough transform or mathematical morphology based methods [29], however these approaches show limitations in cases of dense populations with several adjacent objects. On the other hand MPP models can handle these phenomena more efficiently, through jointly describing individual objects by various data terms, and using information from entity interactions by prior geometric constraints [30]. Although the computational complexity of MPP optimization may mean bottleneck for some applications, various efficient techniques have recently been proposed to speed up the energy minimization process, such as the Multiple Birth and Death (MBD) [23] algorithm or the parallel Reversible-Jump Markov Chain Monte Carlo (RJMCMC) sampling process [31].

However, conventional MPP models offer limited options for hierarchical scene modeling, since they usually exploit pairwise object interactions, which are defined on fixed symmetric object neighborhoods. In a traffic situation we often find several groups of regularly aligned vehicles, but we must also deal with junctions or skewed parking places next to the roads, where many differently oriented cars appear close to each other. In addition, the coherent car groups may have thin, elongated shapes, therefore concentric neighborhoods are less efficient. Some earlier attempts have already been conducted to introduce hierarchical contextual models in the MPP framework. In [32] the relation between objects and object parts has been modeled as a relationship of parent and child objects. Here we need a different approach, since instead of object encapsulation we should give probabilistic models for various object grouping constraints.

For the above reason, we propose a new Two-Level MPP (L^2MPP) model, which partitions the complete vehicle population into vehicle groups, called *traffic segments*, and extracts the vehicles and the optimal segments simultaneously by a joint energy minimization process. While object interactions within the same segment realize conventional non-overlapping or alignment constraints [33], the key novelty of L^2MPP is that we introduce inter layer object – group interaction terms which can prescribe different geometric constraints within different object groups, implementing adaptive object neighborhoods. Features exploited in the recognition process are directly derived from the classification of the Lidar point cloud in 3-D. However, to keep the computational time tractable, the optimization of the inverse problem is performed in 2-D, following a ground projection of the previously obtained class labels. During the projection of the Lidar point cloud to the ground (i.e. a regular image), we do not interpolate pixel values with missing data, avoiding artifacts of data interpolation.

In our model, the processed Lidar scans are considered as 3-D scans of the cities, and we extract local snapshots from the traffic flow, with performing location and orientation based contextual classification of the vehicles. In this way, we can obtain robust information about the number and density of objects in different road lanes, crossroads and parking areas. Our extracted descriptors can contribute to statistical loading analysis of roads and main junctions in different day parts and seasons, completeness study of regular and ad-hoc parking areas, or detecting vehicles with outlier positioning among regularly aligned objects.

TABLE I
PARAMETERS ASSOCIATED TO A POINT p OF THE INPUT CLOUD $\mathcal{L}$ PROVIDED BY A DISCRETE RETURN (DR) LIDAR SENSOR.

Parameter	Domain	Description
x_p, y_p, z_p	$\mathbb{R}^3$	coordinates of the 3-D geometric location of the point p
g_p	[0,255]	calibrated intensity value associated to the point p
η_p	$\{1, 2, 3, 4\}$	total number of range measurements (echos) of the laser pulse yielding p
r_p	$\{1, 2, 3, 4\}$	index (ordinary number) of the echo associated to point p

The workflow and dataflow charts of the proposed method are displayed in Fig. 1 and Fig. 2 respectively. In Sec. II we describe the point cloud classification and ground projection steps. We introduce the proposed L^2MPP model in Sec. III, and the corresponding energy optimization algorithm in Sec. IV. In the experimental part (Sec. V) we discuss first the parameter settings, thereafter qualitative and quantitative results are provided using different group-, object- and pixel-level evaluation metrics. We validate the proposed model on a data set of 1009 vehicles from seven different urban regions, and compare our results to four previous approaches. Finally, concluding remarks are given in Sec. VI. This article extends our corresponding conference papers [24], [34] with significant new model elements, including an improved classification model, various new data based and prior features and generalized grouping constraints for curved roads.

II. CLASSIFICATION OF AERIAL POINT CLOUDS

The first step of the proposed workflow is point cloud classification, as displayed in Fig. 1. Similarly to [35], we have developed an energy minimization based contextual point cloud segmentation method. However, while [35] deals with macro area classification, marking vehicles as part of the clutter regions, our approach also focuses on the accurate discrimination of the vehicle class from other areas.

The input of the proposed framework is a point cloud $\mathcal{L}$ provided by a Discrete Return (DR) airborne Lidar system. Let us assume that the cloud consists of l points: $\mathcal{L} = \{p_1, \ldots, p_l\}$, where each point, $p \in \mathcal{L}$, is associated to six parameters, as listed in Table I. The geometric position coordinates (x_p, y_p, z_p) are available in a local Euclidean coordinate system, which is adjusted to the WGS 84 datum surface. In addition, each point has a calibrated intensity value g_p. As indicated earlier, the DR Lidar system may capture up to four range measurements (echos) for each laser pulse. This information is encoded in the point could by adding two addition parameters to each point: η_p marks the total number of captured echos from the pulse yielding p, and $r_p(\leq \eta_p)$ is the reflection index corresponding to p within the echos of the

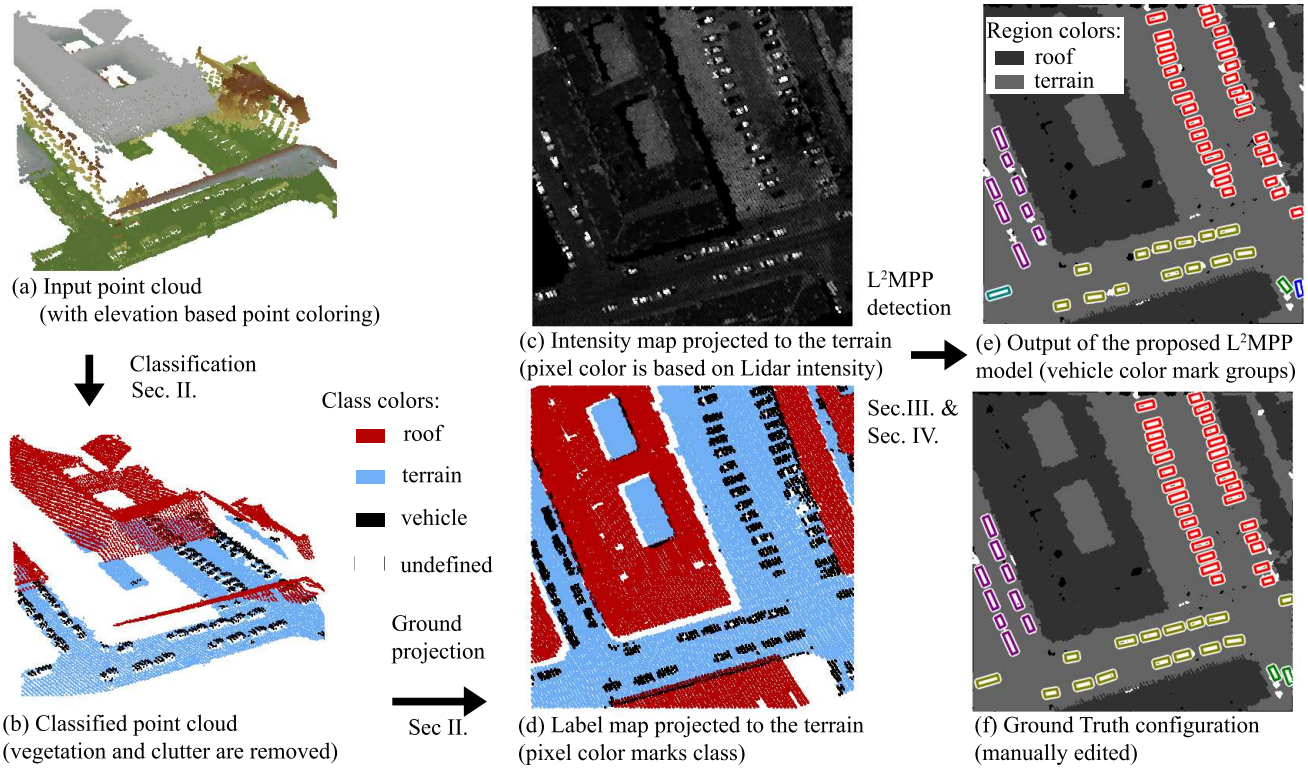


Fig. 1. Workflow of the point cloud filtering, classification and projection steps. Note that for easier visualization, we have distinguished pixels of roof (red) and ground (blue) in the projected label map (Fig. (c)), but during the vehicle extraction process, we consider them as part of a unified background class.

same laser pulse. If $r_p = \eta_p$, we say that p corresponds to a *last return*, otherwise to an *intermediate return*.

Let us denote by $\mathcal{V}_\epsilon(p)$ the ϵ neighborhood of p:

$$\mathcal{V}_\epsilon(p) = \{q \in \mathcal{L} : ||q - p|| < \epsilon\}, \tag{1}$$

where $||q - p||$ marks the Euclidean distance of points q and p, and the ϵ threshold parameter was set as $\epsilon = \sqrt{\frac{1}{2\rho}}$, where ρ is the point density of the scan measured in points/m^2. For efficient neighborhood calculation, we need to divide the point cloud into smaller parts by making a nonuniform subdivision of the 3-D space using a *k*-d tree data structure.

In our classification model, we distinguish *terrain*, *vegetation*, *roof*, *vehicle* and *clutter* regions, and accordingly we denote by $\xi(p)$ the class label assigned to a given point p. The clutter class contains sparse point cloud regions, which mainly correspond to vertical structures such as facades and lampposts, or thin objects, like power lines.

To classify the point cloud, we define for each class ξ a $\mu_\xi(p) \in [0, 1]$ inverse membership (or energy) function, which evaluates the hypothesis that $p \in \mathcal{L}$ belongs to the ξ class, marking high quality matches with lower μ values. For deriving the membership terms we use ζ sigmoid functions, which can be considered as *soft thresholds* [36]:

$$\zeta(x, \tau, m) = \frac{1}{1 + \exp(-m \cdot (x - \tau))}. \tag{2}$$

where $x \in \mathbb{R}$ is a scalar valued fitness descriptor evaluating the match between x and a selected point cloud class; τ is a soft *upper* threshold corresponding to x with respect to the class, and m is a steepness parameter used for normalization. If we need to apply a *lower* threshold constraint for a given feature, we simply need to reflect the sigmoid function to the $y = 0.5$ line, i.e. using $(1 - \zeta(x, \tau, m))$ as class energy function. In this way parameter tuning for the different classes is straightforward, if the evidence of class membership monotonously increases or decreases as a function of the x feature.

Based on the membership terms, we define an E energy function on the space of the possible global point cloud labellings, which uses the Potts smoothness term favoring similar labels for close points [35]:

$$E(\{\xi(p)|p \in \mathcal{L}\}) = \sum_{p \in \mathcal{L}} \mu_{\xi(p)}(p) + \sum_{p \in \mathcal{L}} \sum_{r \in \mathcal{V}_\epsilon(p)} \kappa \cdot \mathbb{I}\{\xi(p) \neq \xi(r)\} \tag{3}$$

where $\kappa > 0$ is the weight of the interaction term and $\mathbb{I}\{.\}$ is an indicator function: $\mathbb{I}\{\text{true}\} = 1$, $\mathbb{I}\{\text{false}\} = 0$.

We continue with the definition of the class membership functions. The first step is *terrain* modeling. Planar *ground* models are frequently adopted in the literature relying on robust plane estimation methods such as RANSAC, however, they are less efficient in cases of significant elevation differences within the observed terrain parts. In these cases bottom parts of the cars can be cut off by the estimated ground plane, or the objects may drift over the ground. *Instead*, we apply a cell based locally adaptive terrain modeling approach [37]. First, we fit a regular 2-D grid with $W_S = 1m$ rectangle width (i.e. grid distance) onto the *horizontal* $P_{z=0}$ plane of the point cloud's Euclidean coordinate system. We assign each $p \in \mathcal{P}$ point to the corresponding cell, which contains the

projection of p to $P_{z=0}$. We mark the cells as *terrain candidate cells* where the differences of the observed maximal and minimal z_p point elevation values are lower than 50cm, which condition admits up to 26° ground slope within a cell. Next, for obtaining a local Digital Terrain Model (DTM), we calculate for the previously marked terrain candidate cells the average of the included point *elevation* coordinates. To eliminate outlier values in the DTM resulted by e.g. flat car roofs, we apply a median filter on the elevation map, and interpolate the remaining cell elevation values from the neighboring *terrain* regions. As the DTM is ready, we calculate for each point p its distance from the terrain model T_p: $d_p^T = \mathrm{dist}(p, T_p)$. For real ground points we expect low height values, therefore we determine the class energy function by soft-thresholding the d_p^T levels:

$$\mu_{\text{terrain}}(p) = \zeta\left(d_p^T, \tau_{\text{ter}}, m_{\text{ter}}\right). \quad (4)$$

Here τ_{ter} is an *upper* height threshold for ground points, which depends on the geometric accuracy of the Lidar data and m_{ter} is a normalizing parameter. We set these factors in a supervised way by training regions, since they highly depend on the noise level and point density of the measurement. Note that using our proposed terrain modeling approach, we may classify the top of large flat roofs as local ground, which enables us to detect vehicles on roof top parking places.

For detecting the *vegetation*, we analyzed the return (echo) numbers of the points. Typically, in regions covered by trees and bushes we can observe multiple laser returns, i.e. $\eta_p - r_p > 0$ holds for vegetation points. Thus for the $\eta_p - r_p$ difference value we can apply 0.5 as soft *lower* threshold to obtain the vegetation class' energy term:

$$\mu_{\text{vegetation}}(p) = 1 - \zeta\left(\eta_p - r_p, 0.5, m_{\text{veg}}\right). \quad (5)$$

Note that multiple laser returns are also present at the edges of buildings, but these regions can mostly be filtered out by the smoothness term of the model.

In *clutter regions*, which are typically formed by reflections from walls in aerial Lidar scans, we expect at most a few ($\tau_{\mathcal{V}}$) neighbors around each point in the cloud:

$$\mu_{\text{clutter}}(p) = \zeta\left(|\mathcal{V}_\epsilon(p)|, \tau_{\mathcal{V}}, m_{\mathcal{V}}\right). \quad (6)$$

As $\tau_{\mathcal{V}}$ soft threshold, we used 30% of the average point density of the point cloud in an $\epsilon \times \epsilon$ vertical column.

Regarding the *roof* class, we assume that the d_p^T height parameter of the point exceeds a τ_{roof} value, and the roof points form dense regions, so that $|\mathcal{V}_\epsilon(p)| > \tau_{\mathcal{V}}$. The energy subterms of these two soft *lower* thresholding constraints are joined with the maximum (i.e. logical AND) operator to obtain the roof class energy:

$$\mu_{\text{roof}}(p) = \max\left(1-\zeta\left(d_p^T, \tau_{\text{roof}}, m_{\text{roof}}\right), 1-\zeta\left(|\mathcal{V}_\epsilon(p)|, \tau_{\mathcal{V}}, m_{\mathcal{V}}\right)\right)$$

Finally, for points corresponding to vehicles we prescribe three soft constraints using the negation of three previously defined terms. We expect that the d_p^T point elevation w.r.t. the local terrain part is between the maximal accepted ground height (τ_{ter}) and the minimal roof height value (τ_{roof}), while the given point should correspond to the last reflection from the corresponding direction:

$$\mu_{\text{vehicle}}(p) = \max\Big(1 - \zeta\left(d_p^T, \tau_{\text{ter}}, m_{\text{ter}}\right), \zeta\left(d_p^T, \tau_{\text{roof}}, m_{\text{roof}}\right), \zeta\left(\eta_p - r_p, 0.5, m_{\text{veg}}\right)\Big) \quad (7)$$

By constructing all the class membership functions, the global energy formula of (3) is completely defined. For the minimum of (3) we can get an efficient approximation by graph-cut based techniques [38], a sample result is shown in Fig. 1(b).

After the 3-D classification process, we stretch a 2-D pixel lattice S (i.e. an image) onto the terrain model, where $s \in S$ denotes a single pixel. Next, we project each Lidar point to this lattice, which has a label of ground, vehicle or building roof, and create a 2-D class label map and an intensity map. The label of pixel s, $\nu(s) \in \{\text{vehicle}, \text{background}, \text{undefined}\}$, is chosen by a majority voting from the $\mu(.)$ labels of points projected to s. Here the union of roof and ground labels form the background class, while $\nu(s) = \text{undefined}$ if no point corresponds to s after projection. We also assign to each pixel s an intensity $g(s)$, which is 0, if $\nu(s) = \text{undefined}$, otherwise we take the average intensity of points projected to s. For point clouds with 8 points/m^2 density, we used a cell side length of 30 cm in S, which means a three times larger grid resolution than the one adopted for terrain modeling. With this choice we assign in average 0.7 points to a pixel, so that information loss due to overlapping point projections will be limited.

In the following part of the algorithm, we purely work on the previously extracted label and intensity images. The detection is mainly based on the label map, but additional evidences are extracted from the intensity image, where several cars appear as salient bright blobs due to their shiny surfaces.

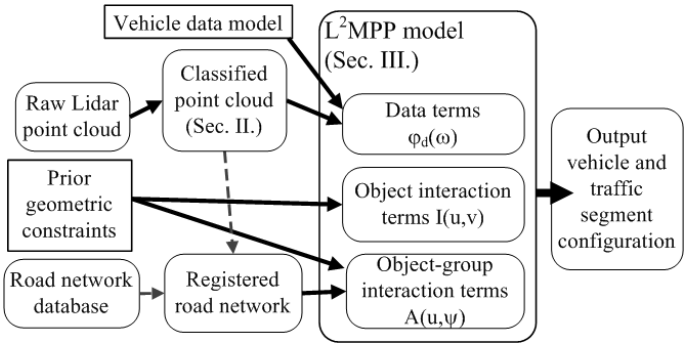


Fig. 2. A dataflow model of the proposed system.

III. L^2-MARKED POINT PROCESS MODEL

The inputs of this step are the label and intensity maps over the pixel lattice S, which were extracted in the previous section (see Fig. 1(c) and (d)). We assume that each vehicle can be approximated from top view by a rectangle, which we aim to extract by the following model. A vehicle candidate u is described by five parameters: c_x and c_y coordinates of the center pixel $c \in S$, side lengths e_L, e_l and orientation $\theta \in [-90^\circ, +90^\circ]$ (Fig. 3(c)). Note that with replacing the rectangle shapes for parallelograms, the "shearing effect" of moving vehicles may also be modeled [1].

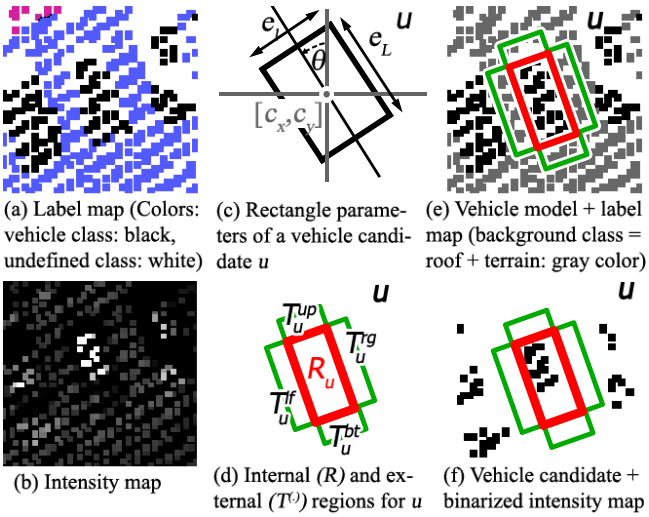


Fig. 3. Demonstration of the (a)-(b) input maps (c) object rectangle parameters and (d)-(f) data term calculation process

Let $\mathcal{H}$ be the space of u objects. We define a neighborhood relation $\sim$ in $\mathcal{H}$: $u \sim v$ iff the distance of the object centers is smaller than a threshold. We describe the scene by a Two-level Marked Point Process (L^2MPP) model: a global configuration ω is a set of k traffic segments, $\omega = \{\psi_1, \ldots, \psi_k\}$, where each traffic segment ψ_i $(i = 1 \ldots k)$ is a configuration of n_i vehicles, $\psi_i = \{u_1^i, \ldots, u_{n_i}^i\} \in \mathcal{H}^{n_i}$. Here we prescribe that $\psi_i \cap \psi_j = \emptyset$ for $i \neq j$, while the k set number and $n_1, \ldots, n_k$ set cardinality values may be arbitrary (and initially unknown) integers. We mark with $u \prec \omega$ if u belongs to any ψ in ω, i.e. $\exists \psi_i \in \omega : u \in \psi_i$. Ω denotes the space of all the possible ω global configurations, and is defined as follows:

$$\Omega = \cup_{k=0}^{\infty} \left\{ \{\psi_1, \ldots, \psi_k\} \in [\cup_{n=1}^{\infty} \Psi_n]^k \right\}$$
$$\text{where } \Psi_n = \{\{u_1, \ldots, u_n\} \in \mathcal{H}^n\}. \tag{8}$$

The above formula expresses that a configuration may consist of any number of traffic segments, and each segment can contain an arbitrary number of vehicles.

Next, following an inverse modeling approach, an energy function $\Phi(\omega)$ is defined, which can evaluate each $\omega \in \Omega$ configuration based on the observed data and prior knowledge. The above neighborhood-energies are constructed by fusing various data terms and prior terms, as it will be introduced in the following subsections in details. Therefore the energy function can be decomposed into a data term and a prior term: $\Phi(\omega) = \Phi_d(\omega) + \Phi_p(\omega)$, and the optimal ω is obtained by minimizing $\Phi(\omega)$.

A. Data-dependent energy terms

Data terms evaluate the proposed vehicle candidates (i.e. the $u = \{c_x, c_y, e_L, e_l, \theta\}$ rectangles) based on the input label- or intensity maps, but independently of other objects of the population. The data modeling process consists of two steps. *First*, we define different $f(u) : \mathcal{H} \to \mathbb{R}$ features which evaluate a vehicle hypothesis for u in the image, so that 'high' $f(u)$ values correspond to efficient vehicle candidates. In the *second step*, we construct $\varphi_d^f(u)$ *data driven* energy subterms for each feature f, by attempting to satisfy $\varphi_d^f(u) < 0$ for real objects and $\varphi_d^f(u) > 0$ for false candidates. For this purpose, we project the feature domain to $[-1, 1]$ with a monotonously decreasing function [23]: $\varphi_d^f(u) = \mathcal{Q}\big(f(u), d_0^f\big)$, where

$$\mathcal{Q}(x, d_0) = \begin{cases} \left(1 - \frac{x}{d_0}\right), & \text{if } \; x < d_0 \\ \exp\left(-\frac{x - d_0}{0.1}\right) - 1, & \text{if } \; x \geq d_0. \end{cases} \tag{9}$$

Observe that the $\mathcal{Q}$ function has a key parameter, d_0, which is the object acceptance threshold for feature x.

We used four different data-based features, which are demonstrated in Fig. 3. Let us denote by $R_u \subset S$ the pixels of the image lattice lying inside the u vehicle candidate's rectangle, and by T_u^{up}, T_u^{bt}, T_u^{lt}, and T_u^{rg} the upper, bottom, left and right object neighborhood regions, respectively (see Fig. 3(d)). The feature definitions are listed in the following paragraphs.

The *vehicle evidence* feature $f^{\mathrm{ve}}(u)$ expresses that we expect several pixels classified as vehicle within R_u:

$$f^{\mathrm{ve}}(u) = \frac{1}{|R_u|} \sum_{s \in R_u} \mathbb{I}\{\nu(s) = \mathrm{vehicle}\}, \tag{10}$$

where $|R_u|$ denotes the cardinality of R_u, and $\mathbb{I}\{.\}$ marks again an indicator function.

The *external background* feature $f^{\mathrm{eb}}(u)$ measures if the vehicle candidate is surrounded by background regions:

$$f^{\mathrm{eb}}(u) = \underset{i \in \{\mathrm{up,bt,lt,rg}\}}{\min 2\mathrm{nd}} \left(\frac{1}{|T_u^i|} \sum_{s \in T_u^i} \mathbb{I}\{\nu(s) = \mathrm{backgr.}\} \right), \tag{11}$$

where the $\min 2\mathrm{nd}$ operator returns the second smallest element from the background filling ratios of the four neighboring regions, thus we also accept vehicles which connect with at most one side to other vehicles or undefined regions.

The *internal background* feature $f^{\mathrm{ib}}(u)$ prescribes that within R_u only very few background pixels may occur:

$$f^{\mathrm{ib}}(u) = \frac{1}{|R_u|} \sum_{s \in R_u} 1 - \mathbb{I}\{\nu(s) = \mathrm{backgr.}\}. \tag{12}$$

Calculation of the f^{ve}, f^{eb} and f^{ib} features can be followed in Fig. 3(e).

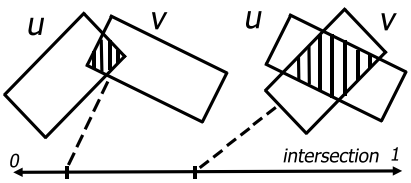
Finally, the *intensity* feature provides additional evidence for image parts containing high intensity regions (see Fig. 3(b) and (f)).

$$f^{\mathrm{it}}(u) = \frac{1}{|R_u|} \sum_{s \in R_u} \mathbb{I}\{g(s) > T_g\}, \tag{13}$$

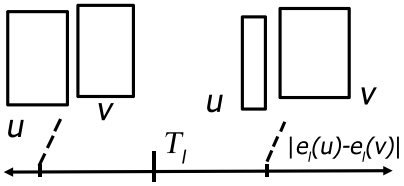
where T_g is an intensity threshold. As Fig. 1(c) shows many vehicles appear as bright blobs in the asphalt, which fact makes the feature relevant to support the detection process.

After the feature definitions, the data terms $\varphi_d^{\mathrm{it}}(u)$, $\varphi_d^{\mathrm{ve}}(u)$, $\varphi_d^{\mathrm{ib}}(u)$, $\varphi_d^{\mathrm{eb}}(u)$ can be calculated with the $\mathcal{Q}$ function by appropriately fixing the corresponding d_0^f parameters for each feature. We set the parameters based on manually annotated training data obtained by using a ground truth generation tool, which will be described later in Sec. V-C.

Once we obtained the subterms, the joint data energy of



(a) Overlapping feature



(b) Width uniformity feature

Fig. 4. Demonstration of the used pairwise interaction constraints

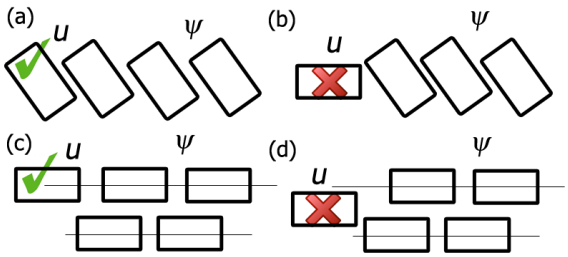


Fig. 5. Favored (√) and penalized (×) sub-configurations within a traffic segment

object u is derived as

$$\varphi_d(u) = \max(\min(\varphi_d^{\text{it}}(u), \varphi_d^{\text{ve}}(u)), \varphi_d^{\text{eb}}(u), \varphi_d^{\text{ib}}(u)). \quad (14)$$

Here the min and max operators are equivalent to the logical OR resp. AND operations for the different feature constraints in the negative fitness domain. We do not prescribe simultaneously the *vehicle evidence* and *intensity* constraints, since usually not all vehicles appear as bright blobs in the intensity map. The data term of the ω configuration is obtained as the sum of the individual object energies: $\Phi_d(\omega) = \sum_{u \prec \omega} \varphi_d(u)$.

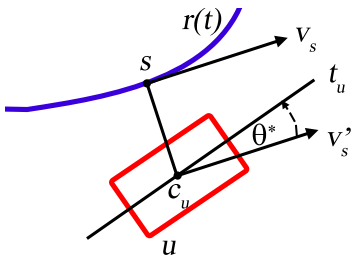


Fig. 6. Roadside-dependent orientation calculation. c_u: center pixel of object u, t_u main axis of u, $r(t)$: roadside's parametric curve, $s \in r(t)$: closest point of $r(t)$ to c_u, v_s roadside tangent vector at s. θ^*: angle between v_s and t_u

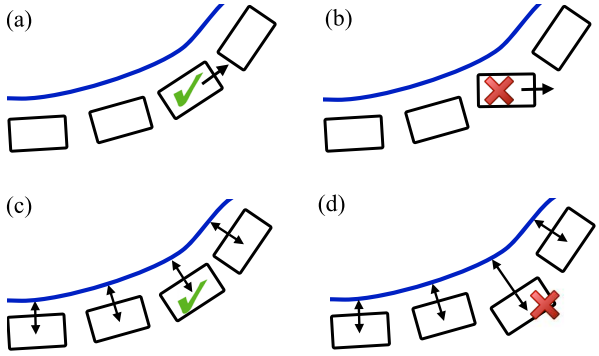


Fig. 7. Roadside-dependent prior orientation terms within a traffic segm.

B. Prior terms

The prior terms encode geometric knowledge about the expected ω vehicle populations. The prior configuration energy is decomposed into two main parts:

$$\Phi_p(\omega) = \sum_{\substack{u,v \prec \omega \\ u \sim v}} I(u,v) + \sum_{u \prec \omega, \psi \in \omega} A(u,\psi). \quad (15)$$

Here the $I(u,v)$ terms implement classical pairwise interaction constraints between (spatially) neighboring objects, constructed in a similar manner to various examples from [30]. On one hand, we penalize any overlapping rectangles within the ω configuration (see Fig. 4(a)). On the other hand, to prevent us from merging contacting vehicles into the same object candidate, we penalize rectangles with significantly different width (e_l) parameters in local neighborhoods (Fig. 4(b)):

$$I(u,v) = \frac{\text{Area}\{R_u \cap R_v\}}{\text{Area}\{R_u \cup R_v\}} + \\ + \frac{1}{|\mathcal{N}_u|} \sum_{v \in \mathcal{N}_u} \mathbb{I}\{|e_l(u) - e_l(v)| > T_l\} \quad (16)$$

where $\mathcal{N}_u = \{v \prec \omega : u \sim v\}$ marks the neighborhood of u. We set T_l as the half of the average vehicle width. Our experiments showed that this assumptions did not yield further false detection results, only the width estimation might be slightly inaccurate for very wide vehicles.

On the other hand, the $A(u,\psi)$ terms can describe various constraints between the object group level and the object level of the scene, which can be considered as the main novelty of the proposed L^{2}MPP model.

The object grouping process is based on the relative orientation and positioning of the vehicles close to each other. More specifically, in a straight road, we prescribe that the vehicles of the same traffic segment have similar orientation, and they form regular rows: Fig. 5 shows examples for favored and penalized configurations within a given vehicle group. If we also need to deal with *curved road* parts such as exit ramps or roundabouts, we should notice that the *rows* of corresponding vehicles may follow curved lanes. At this point we utilize the available road network information mentioned in Sec. I-C. For each vehicle, we calculate a relative orientation w.r.t. the local road tangent for the classification. More specifically, as shown in Fig. 6 we calculate the *roadside angle* θ^* as

the angle between the main axis of the vehicle and the tangent of the road curve in the closest contour point to the vehicle center. Positive and negative samples for appropriate alignments within a group are shown Fig. 7.

To define the $A(u,\psi)$ energy components, we introduce an alignment distance term $d_\psi(u) \in [0,1]$, which measures whether a vehicle u is appropriately arranged with respect to a traffic segment ψ. In our model, $d_\psi(u)$ is the average of two subterms. *First*, we take a normalized angle difference between $\theta^*(u)$ and the mean angle θ^*_ψ within ψ: $\min(|\theta^*(u) - \theta^*_\psi|, 45^\circ)/45^\circ$ (see Fig. 5(a),(b) and Fig. 7(a),(b)). *Second*, we calculate a distance term between the center of object u and the lane orientation curve, normalized by the expected average lane width in the scene. By considering straight road segments only, the lane orientation curve is obtained as lines fit to the object centers of the group (Fig. 5(c),(d)). Otherwise, the reference lane orientation curve is the local part of the used road network (Fig. 7(c),(d)).

After defining the $d_\psi(u)$ distance metric, we construct the group alignment energy term. For prescribing spatially connected traffic segments, we use a constant high difference factor, if u has no neighbors within ψ w.r.t. relation $\sim$. Thus we derive a modified distance:

$$\hat{d}_\psi(u) = \begin{cases} 1 & \text{if } \nexists v \in \psi \backslash \{u\} : u \sim v \\ d_\psi(u) & \text{otherwise} \end{cases} \tag{17}$$

We define the $A(u,\psi)$ arrangement term of (15) by discriminating three cases. First, we slightly penalize vehicle groups which only contain a single vehicle. Second, between a segment ψ and an included object $u \in \psi$ we penalize large $\hat{d}_\psi(u)$ distance values. Third, we also penalize, if u does not belong to ψ, although the $\hat{d}_\psi(u)$ distance is *low*. The above three constraints are formulated as follows:

$$A(u,\psi) = \begin{cases} c & \text{if } \psi = \{u\} \\ \hat{d}_\psi(u) & \text{if } u \in \psi \\ 1 - \hat{d}_\psi(u) & \text{if } u \notin \psi \end{cases} \tag{18}$$

where $0 < c \ll 1$.

IV. Optimization

MPP energy functions are optimized in the literature with iterative stochastic algorithms, most frequently with the Reversible Jump Markov Chain Monte Carlo (RJMCMC) sampler [39] or the Multiple Birth and Death Dynamic technique (MBD) [23]. In most previous RJMCMC based solutions, each iteration of the relaxation consists in perturbing one or a couple of objects with various kernels such as birth, death, translation, rotation or dilation. Here experiments show that the rejection rate, especially for the birth move, may induce a heavy computation time. Besides, one should decrease the temperature slowly, because at low temperature, it is difficult to add objects to the population. On the other hand, MBD [23] evolves the population of objects by alternating purely stochastic object generation (*birth*) and removal (*death*) steps, in a Simulated Annealing (SA) framework. In contrast to the above mentioned RJMCMC implementations, each birth step of MBD consists of adding several random objects to the current configuration, and there is no rejection during the birth step, therefore high energetic objects can still be added independently of the temperature parameter. Due to these properties, in several remote sensing tasks notable gain has been reported in optimization speed versus RJMCMC [23], [28]. On the other hand, parallel sampling in MBD implementations is less straightforward than regarding the RJMCMC relaxation [31].

We have chosen for our method the extension of the MBD algorithm, as an efficient trade-off between performance and processing speed. As MBD has been designed for single layer MPP models, the main task was here to include the group assignment and the object re-grouping issues within the original framework. More specifically, after each *birth* step, the generated object should be assigned to a new, or an existing group. Then, after the *death* procedure, we execute a new step, called *Group re-arrangement*, which may re-direct some objects to neighboring segments based on data based and alignment features.

The steps of the modified, two-level MBD algorithm are as follows:

Initialization: start with empty population $\omega = \emptyset$, set the birth rate b_0, initialize the inverse temperature parameter $\beta = \beta_0$ and the discretization step $\delta = \delta_0$.

Main program: alternate the following three steps:

• *Birth step*: Visit all pixels on the image lattice S one after another. At each pixel s, with probability δb_0, generate a new object u with center s and random e_L, e_l and θ parameters. For each new object u, with a probability

$$p^0_u = \mathbb{I}\{\omega = \emptyset\} + \mathbb{I}\{\omega \neq \emptyset\} \cdot \min_{\psi_j \in \omega} \hat{d}_{\psi_j}(u), \tag{19}$$

generate a new ψ empty traffic segment, add u to ψ and ψ to ω. Otherwise, add u to an existing traffic segment $\psi_i \in \omega$ with a probability

$$p^i_u = \frac{(1 - \hat{d}_{\psi_i}(u))}{\sum_{\psi_j \in \omega}(1 - \hat{d}_{\psi_j}(u))}. \tag{20}$$

• *Death step*: Consider the actual configuration of all objects within ω and sort it by decreasing values depending on $\varphi_d(u) + A(u,\psi)\big|_{u\in\psi}$. For each object u taken in this order, compute $\Delta\Phi_\omega(u) = \Phi(\omega/\{u\}) - \Phi(\omega)$, derive the *death rate* $p^d_\omega(u)$ as

$$p^d_\omega(u) = \Gamma(\Delta\Phi_\omega(u)) = \frac{\delta \exp(-\beta \cdot \Delta\Phi_\omega(u))}{1 + \delta \exp(-\beta \cdot \Delta\Phi_\omega(u))}, \tag{21}$$

and delete object u with probability $p^d_\omega(u)$. Remove empty population segments from ω, if they appear.

• *Group re-arrangement*: Consider the objects of the current ω population, one after another. For each object u of segment ψ we propose an alternative object u', so that the geometric parameters of u' are derived from the parameters of u by adding zero mean Gaussian random values. The next step is selecting a group candidate for u'. For this reason, we randomly choose a v object from the proximity neighborhood of u ($v \in \mathcal{N}_u(\omega)$), and assign u' to the group of v, denoted by ψ'. Then, we estimate the energy cost of exchanging $u \in \psi$

TABLE II
CATEGORIZATION OF THE DATA SETS BY DIFFERENT CONTENT FEATURES, WITH ALSO GIVING THE COVERED AREA AND POINT NUMBER

Feature / Data set	#1	#2	#3	#4	#5	#6	#7
Main road traffic	×		×	×	×		×
Roadside parking	×	×	×		×		×
Parking square	×			×		×	
Curved Road	×				×		
Cluttered traffic	×	×				×	×
Area in 10^{-3}km^2	46	65	39	47	37	39	46
Point num $\cdot 10^4$	45	33	35	38	27	36	36

× marks the features of the different test sets

to $u' \in \psi'$:

$$\Delta\varphi(\omega, u, u') = \varphi_d(u') - \varphi_d(u) + I(u', \omega \backslash \{u\}) - \\ - I(u, \omega) + A(u', \psi') - A(u, \psi) \qquad (22)$$

The *object exchange rate* is calculated using the $\Gamma(.)$ function defined by (21):

$$p_\omega^{\mathrm{e}}(u, u') = \Gamma\Big(\Delta\varphi(\omega, u, u')\Big) \qquad (23)$$

Finally with a probability $p_\omega^{\mathrm{e}}(u, u')$, we replace u with u'.

Convergence test: if the process has not converged yet, increase β and decrease δ with a geometric scheme, and go back to the birth step.

Although the two-level MBD algorithm cannot theoretically guarantee to reach the global minimum of the MPP energy function, it proved to be practically efficient for our addressed problem, which fact will be demonstrated in the next experimental section.

V. EXPERIMENTS AND EVALUATION

We evaluated our method in seven aerial Lidar data sets (provided by Astrium GEO-Inf. Services Hungary), which are captured above dense urban areas of Budapest, Hungary. The collected point clouds have an average point density of 8 points/m^2 considering the last returns. Various traffic situations can be observed in the used data collection, such as main road traffic, roadside parking, parking in squared lots, or cluttered scenarios. A subjective human classification of the test sets by typical events and road configurations is given in Table II. As shown here, the individual data sets cover regions of 0.037 to 0.065 km^2, and the total number of points (including both intermediate and last returns) varies between 270K and 450K. The first set consists of different point cloud parts covering smaller areas, while the remaining sets correspond to larger connected regions. The whole test data collection contains in aggregate 1009 vehicles.

A. Parameter settings

We can divide the parameters of the proposed L^{2}MPP technique into *four* groups corresponding to point cloud *classification*, *data*-based vehicle models, *prior* configuration-level constraints and *optimization*.

The parameters of the *classification*, the *data* and the *prior* terms are set based on manually labeled point cloud regions and training objects, respectively. Most of the data-dependent parameters are related to physical circumstances of the measurement, such as altitude and speed of the airplane, frequency of scanning, measurement noise and point cloud density. We have observed that using similar settings in a given Lidar measurement platform, we do not need to re-calibrate the model parameters for each test set. The later phenomenon is a significant advantage of processing Lidar data, rather than optical images where the parametric models should also be adapted to the outside illumination conditions.

Finally, to set the *optimization* parameters, we followed the guidelines provided in [23] and used $b_0 = 5 \cdot 10^{-6}$, $\delta_0 = 10000$, $\beta_0 = 20$ and geometric cooling factors $1/0.96$.

B. Reference Methods

For comparative evaluation, we have first selected three state-of-the art techniques of Lidar based vehicle detection. Since vehicle grouping has not been investigated by the considered reference methods, we also compared the proposed two-level L^{2}MPP model to a sequential approach which consist of a vehicle detection step with our single layer MPP model (sMPP, [34]), and the grouping step is performed in the post processing phase. Next we briefly introduce the reference methods.

1) DEM-PCA (D-PCA): This method is a *bottom-up* grid-based algorithm introduced in [16], which consists of three consecutive steps: (1) Height map (or *D*igital *E*levation *M*odel) generation by ground projection of the elevation values in the Lidar point cloud, and missing data interpolation. (2) Vehicle region detection by thresholding the height map followed by morphological connected component extraction. (3) Rectangle fitting to the detected vehicle blobs by *P*rincipal *C*omponent *A*nalysis.

2) h-max: : The method proposed by [18] applies three consecutive steps: geo-tiling for accelerating the data-access, vehicle-top detection by local maximum filtering, and segmentation through marker-controlled watershed transformation. Since the output of [18] is a set of vehicle contours, we calculate the bounding boxes of the obtained vehicle blobs to make the direct comparison with our approach relevant.

3) Floodfill: The third algorithm implements a 3-D connected component analysis on the segmented point cloud. First, the point set is classified using our segmentation algorithm presented in Sec. II. Thereafter, in vehicle-classified regions the individual objects are separated by floodfill propagation. We use here a *k*-d tree subdivison for efficient extraction of the nearest neighbor point, and Euclidean distance constraint for vehicle blob separation.

4) single layer MPP (sMPP): we extract the vehicle configuration by our previously proposed sMPP model [34], which uses similar data terms to the present approach, but instead of the complex two-level grouping strategy of L^{2}MPP, simple pairwise energy terms are applied as soft constraints within the interaction components of the MPP energy function. Since the output of [34] is an unsegmented vehicle population, the

for each $i = 1 \dots \hat{n}$
  select $j \in \{1, \dots, \hat{n}\}$ so that $a(i,j) = 1$
  if $j > n$
    u_i detected object is a False Positive
  elseif $i > m$
    v_j GT object corresponds to a False Negative
  elseif $t(i,j) > r_h$ (used $r_h = 8\%$)
    u_i is a True Positive candidate, and u_i is matched to GT object v_j
  else
    u_i is a False Positive and v_j GT object indicates a False Negative
  endif

Fig. 8. Algorithm of object assignment considering the Ground Truth (GT)

grouping step is performed in post processing, by a floodfill based strategy. Starting from a randomly chosen object, we assign all its spatial neighbors to the same cluster iff the difference between the orientations is lower than a threshold (used $25°$), and recursively repeat the process until all objects receive a group label.

C. Automated evaluation methodology

For accurate Ground Truth (GT) generation, we have developed an accessory program with graphical user interface, which enables us to manually create and edit a GT configuration of rectangles and assign each rectangle to a group by operators. To enable fully automated evaluation, we need to make first a non-ambiguous assignment between the detected vehicles $u_1 \dots u_m$ and the GT object samples $v_1 \dots v_n$. Let us denote by $\hat{n} = \max(m, n)$. First, we calculate a similarity matrix $\mathbf{T} = [t(i,j)]_{\hat{n} \times \hat{n}}$ which contains the normalized intersection area of the object rectangles:

$$t(i,j) = I^*(u_i, v_j) = 2 \cdot \frac{|R_{u_i} \cap R_{v_j}|}{|R_{u_i}| + |R_{v_j}|} \quad \text{if } i \le m,\ j \le n \tag{24}$$

otherwise $t(i,j) = 0$. We use the Hungarian algorithm [40] to find the maximum matching, *i.e.* the maximum utilization of $\mathbf{T}$. We denote by $\mathbf{A} = [a(i,j)]_{\hat{n} \times \hat{n}}$ the assignment obtained by the algorithm, which is a binary matrix where each row and each column contains exactly one match denoted by $a(i,j) = 1$. Thereafter, we classify the objects according to the algorithm presented in Fig. 8 as True Positive (TP), False Positive (FP) or False Negative (FN).

We have performed quantitative evaluation both at object and at pixel levels considering the GT configurations. At object level, we have counted number of the TP, FP and FN samples based on the algorithm of Fig. 8. Then, using the Number of real Vehicles (NumV=TP+FN), the F-rate of the detection (harmonic mean of precision and recall) is calculated. At pixel level, we compare the vehicle silhouette mask to the GT mask, and calculate the F-rate of the match [34].

Regarding the *sMPP* and the proposed L^{2}MPP approaches, we have also measured the correct Group Classification Rate (GR, %) among the true positive samples, considering GT classification of human observers. The GR value is determined by counting the number of correctly grouped vehicles (TG), the number of falsely grouped (but correctly detected) objects (FG), and calculating GR=TG/(TG+FG).

TABLE IV
IMPROVEMENTS OF THE L^{2}MPP TECHNIQUE IN TERMS OF CORRECT VEHICLE GROUPING RATE (GR) VERSUS THE SEQUENTIAL SMPP MODEL. FURTHER NOTATIONS ARE DEFINED IN SEC. V-C.

Set	sMPP			L^{2}MPP		
	TG	FG	**GR**	TG	FG	**GR**
#1	170	13	93%	181	3	98%
#2	53	38	58%	80	11	88%
#3	114	49	70%	158	4	98%
#4	120	37	76%	153	4	97%
#5	64	45	59%	100	9	92%
#6	106	23	82%	126	2	98%
#7	104	38	73%	129	14	90%
All	731	243	75%	927	47	95%

Note: TG+FG is equal to the number of True Positive objects

D. Performance evaluation

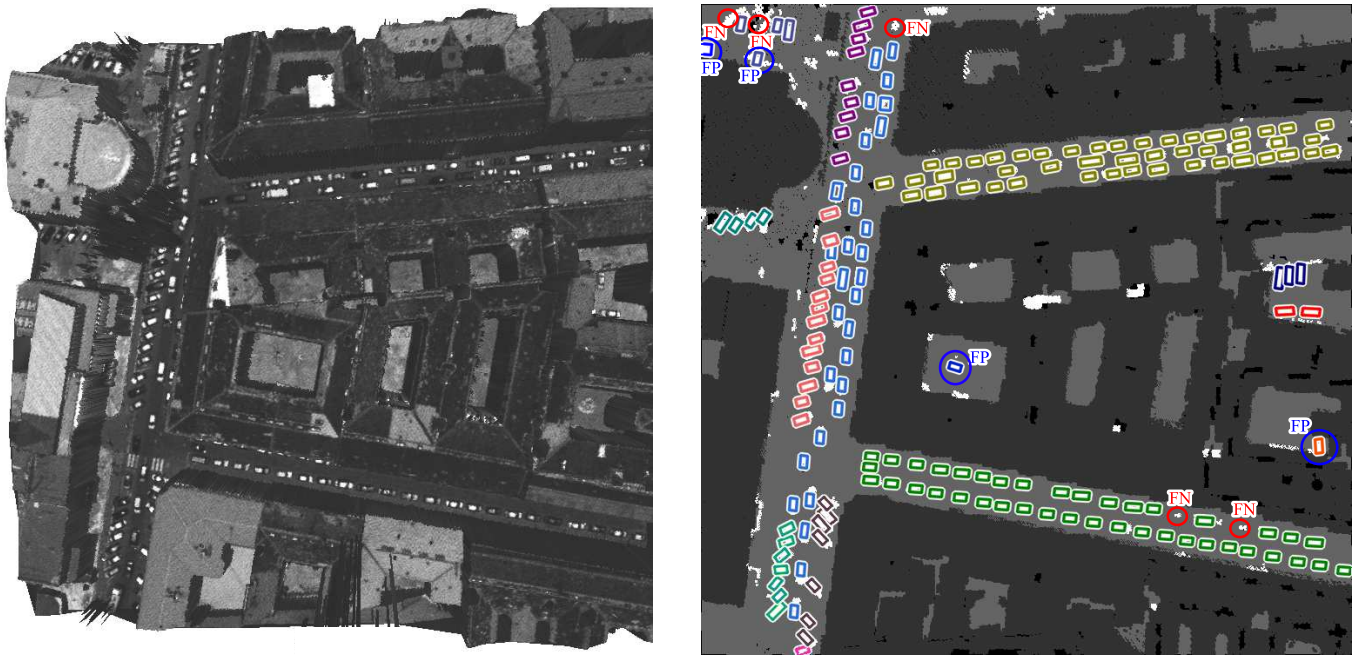
A few qualitative sample results are shown in Fig. 9-13 and the quantitative evaluation is provided in Tables III and IV. In Fig. 9 the complete scene of Data set #3 is displayed with dense traffic and 9 different object groups. We can observe that apart from the few highlighted False Positive (FP) and False Negative (FN) hits, the major part of the vehicles are correctly detected, separated from each other and grouped based on the actual traffic situation. Using the orientation-based grouping constraint the cars parking in a skewed formation can be efficiently distinguished. However, since no car-velocity information is extracted in the proposed model, vehicles parking in parallel to the lanes may be ordered to the traveling cars' traffic segments (see light blue group in Fig. 9). We can also see two False Positives in the central regions of the scene, which are caused by point cloud classification errors. Nevertheless, the five parking cars in a courtyard on the right central part of the image are appropriately detected and aligned by the method. We can also notice that the vehicles parking next to the main top-bottom road part are split into three different groups, which are not spatially connected. Sample parts from the remaining data sets are displayed in Fig. 10.

Although the reference methods were chosen so that they provide complex and valid solutions for the vehicle detection task in general urban environments, we have also observed a number of limitations for each case. Most of the problems with *DEM-PCA* originate from the inaccuracies and discretization artifacts of the estimated elevation maps. In addition, short vegetation or various street objects can corrupt the process since their elevation range is often overlapping with the vehicles' height values. By testing the *h-max* method, we have noticed similar limitations as mentioned by the authors in [18]: in parking areas and cluttered regions, the technique yields inaccurate contours and merges some of the nearby

TABLE III
OBJECT LEVEL AND PIXEL LEVEL F-RATES (IN %) BY THE D-PCA [16], H-MAX [18], FLOODFILL, SMPP [34] AND THE PROPOSED L^2MPP METHODS.

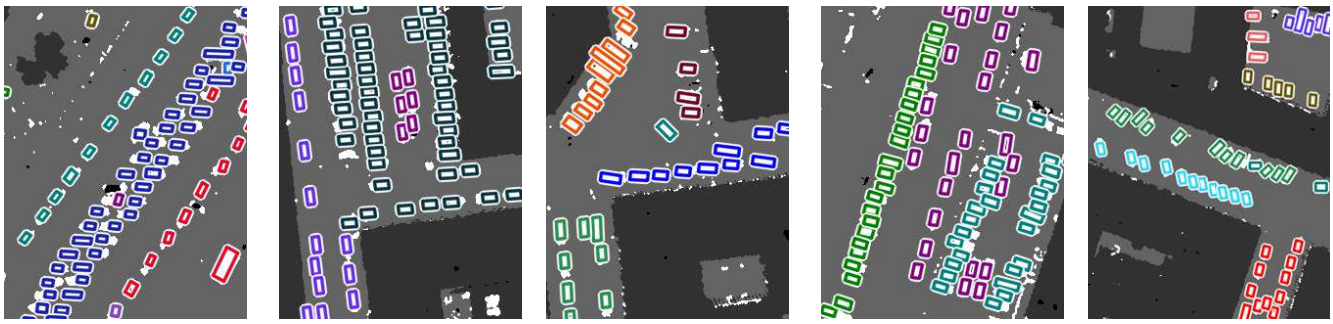
Set	NumV*	Object level F-rate %					Pixel level F-rate%				
		D-PCA	h-max	Floodfill	sMPP	**L^2MPP**	D-PCA	h-max	Floodfill	sMPP	**L^2MPP**
#1	191	78	78	88	97	**97**	63	63	66	81	**82**
#2	94	89	81	80	96	**97**	80	38	60	73	**73**
#3	170	85	87	91	97	**96**	77	76	85	75	**74**
#4	160	68	77	88	97	**97**	61	68	75	80	**89**
#5	110	48	79	92	98	**98**	37	61	82	80	**84**
#6	131	89	81	73	98	**98**	80	70	48	81	**88**
#7	153	80	90	88	93	**93**	60	76	65	74	**88**
All	1009	77	82	86	97	**97**	66	65	71	78	**83**

*NumV = Number of real Vehicles in the test set



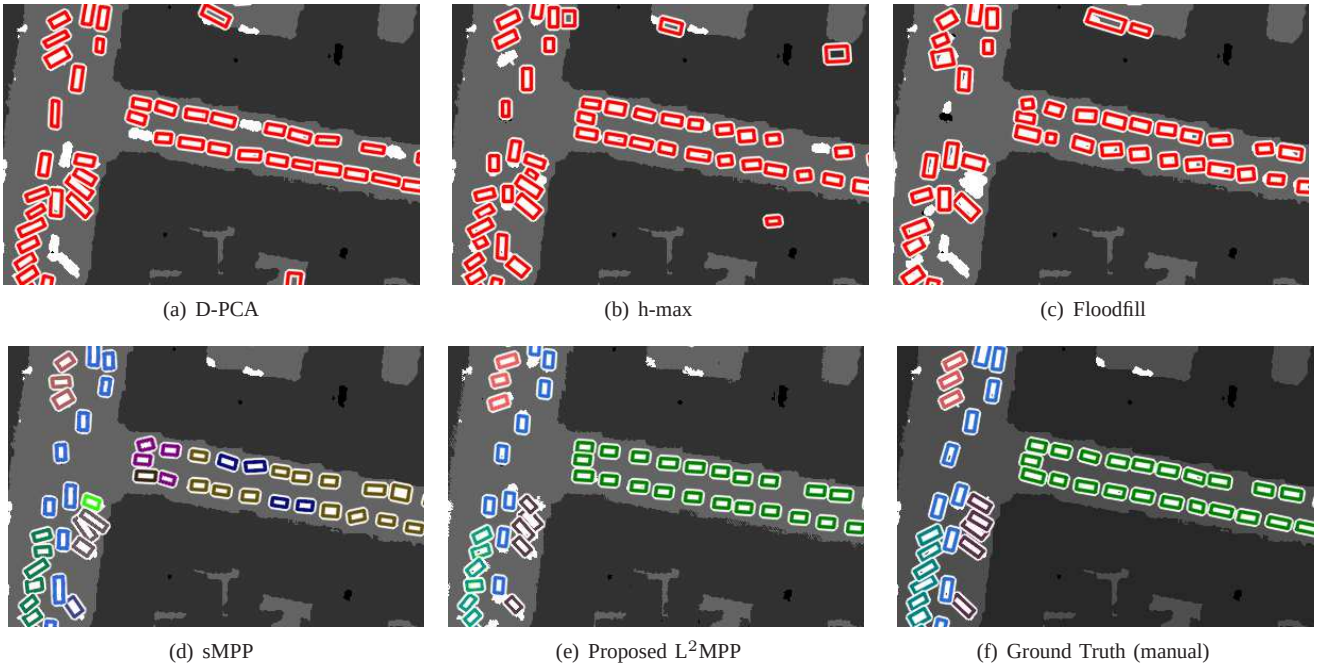
(a) 3-D input Lidar data (mesh visualization with intensity coloring) (b) Detection result showing False Positive (FP) and False Negative (FN) hits

Fig. 9. Demonstration of the result in a large scene part from Data set #3. (a) Input Lidar data visualized as a 3-D triangulated mesh with intensity coloring (note: some vehicles are occluded from this viewpoint) (b) L^2MPP detection result in the 2-D projected plane. Vehicles of different segments are displayed with different colors, background is interpolated for visualization.



(a) Data set #2 (b) Data set #4 (c) Data set #5 (d) Data set #6 (e) Data set #7

Fig. 10. Results on selected regions from the different data sets. Note that a sample from Set #1 is shown in Fig. 13 and Set #3 is displayed in Fig. 9.



(a) D-PCA (b) h-max (c) Floodfill

(d) sMPP (e) Proposed L^{2}MPP (f) Ground Truth (manual)

Fig. 11. Method comparison on a sample part of Fig. 9. Reference methods in the top row do not perform vehicle grouping.

objects, while vegetation causes a number of additional false alarms. Regarding the *Floodfill* algorithm, we observed that 3-D connected component propagation is sensitive to noise due to partial occlusion, and nearby vehicles are often merged together. On the other hand, in the proposed technique the 2-D projection implements already a noise filtering step, and the inverse object description approach of MPP does not request strictly connected components for detecting a vehicle.

Fig. 11 shows a selected segment of the Data set #3, for comparing the output of the reference methods, the proposed model and the manually edited Ground Truth (GT) configuration. Regarding the sMPP, L^{2}MPP and the GT configurations, different vehicle groups are marked with different colors (best viewed in color print), for the three other methods only the vehicle extraction step is investigated. The corresponding numerical object and pixel level evaluation rates (F-rates) are listed in Table III. Both the qualitative and the quantitative results confirm that the proposed L^{2}MPP model surpasses the *D-PCA*, *h-max* and *Floodfill* state-of-the-art techniques at both levels.

The object level performance of the single layer MPP (sMPP) and the proposed model is nearly identical due to the same data energies applied in both cases. However, the pixel level performance of L^{2}MPP is noticeably higher, showing that the prior alignment constraints within the corresponding segments increase the detection accuracy for the noisy data: numerous misaligned or partially extracted vehicle blobs are also shown in Fig. 11(d).

Regarding the Group Classification rate, even a more significant gain is obtained by the proposed L^{2}MPP technique. As listed in Table IV L^{2}MPP outperforms sMPP in the GR factor by 5–30% on the different data sets. Fig. 11 (bottom row) also shows that the segmented population by the two level model is much closer to the human classification than the result of the sequential sMPP approach.

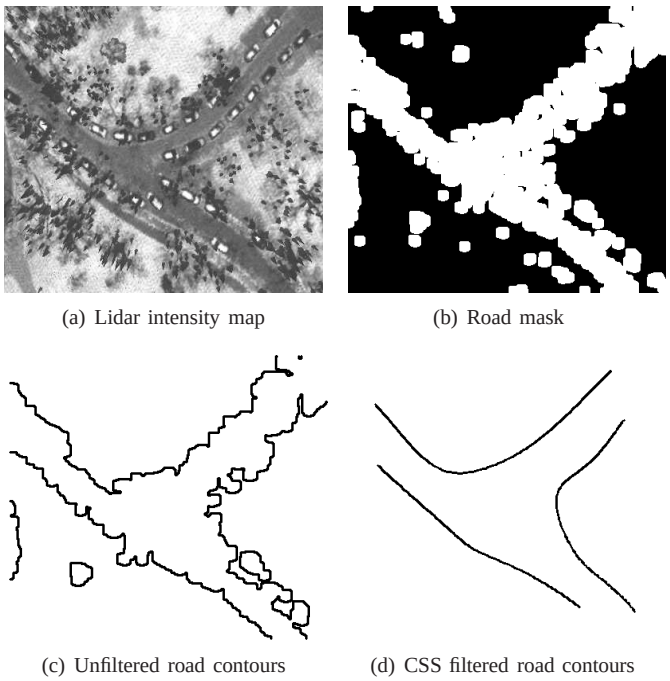
TABLE V
MEASURED COMPUTATIONAL TIME REQUIREMENTS OF THE DIFFERENT METHODS FOR DATA SET #5

Method	Running time
DEM-PCA	57 sec
h-max	55 sec
Floodfill	28 sec
Proposed L^2MPP	53 sec

Using a standard desktop computer and single-thread implementations of the algorithms, we have also measured the running times of the different methods on Data set #5. Although the two-level MBD optimization induces some computational overload, the proposed method is still competitive with most of the reference techniques, and it is only outperformed by Floodfill.

E. Example for road extraction and curved segment analysis

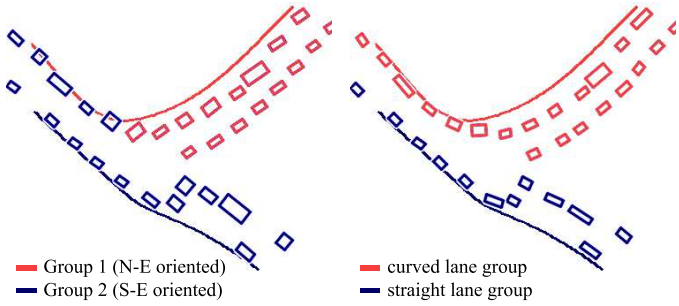
In this subsection, we demonstrate the steps of the proposed algorithm on a challenging data sample with a curved crossroad hidden by dense tree crowns. The input point cloud is shown in Fig. 14(b), and the result of echo number based vegetation removal in Fig. 14(c). In this case, the road network extraction step can be done in an automated way. First, we can observe in the intensity map of Fig. 12(a), that the asphalt regions provide lower intensities than the neighboring natural ground areas. By applying an intensity based thresholding step followed by a morphological closing filter, we can obtain a coarse road mask shown in Fig. 12(b). Since we may find



(a) Lidar intensity map (b) Road mask

(c) Unfiltered road contours (d) CSS filtered road contours

Fig. 12. Intensity based road detection, and contour filtering with the Curvature Scale Space (CSS) technique in a sample



(a) Without curvature constraints [24] (b) Proposed detection results

Fig. 13. Results on a curved road segment, also shown in Fig. 14 and 12.

TABLE VI
RELEVANCE STUDY OF THE DIFFERENT CONFIGURATION ENERGY COMPONENTS ON DATA SET #3. AT PIXEL LEVEL, THE RECALL THE PRECISION AND THE F-RATES ARE ALSO GIVEN

Skipped feature		Obj. level		Pixel level (%)		
		FP	FN	Rec.	Prec.	F-r
1	intensity f^{it}	5	8	65	77	70
2	veh. evid. f^{ve}&f^{it}	57	9	66	57	61
3	internal bg. f^{ib}	26	30	44	42	43
4	external bg. f^{eb}	12	11	57	75	64
5	width-uniform.	5	7	66	78	71
6	– all features used	4	6	67	82	74

vehicles parking on grass in city green areas, we do not restrict the car extraction step to the roads. Instead, we estimate the road contours which will be used for calculating the relative orientation of the vehicles for the traffic segment extraction step. However, since the intensity based road mask is notable noisy here (Fig. 12(b)), we apply for the detected contours (Fig. 12(c)) a robust smoothing process. Here we have filtered the initial contours with the Curvature Scale Space (CSS) technique [41], which yielded the road outlines shown in Fig. 12(d).

The detection result on this road segment is demonstrated in Fig. 13. We compare the model proposed hereby (Fig. 13(b)) to an earlier model version (Fig. 13(a)) introduced in [24], which considers only the parallel alignment constraints for straight roads, but does not use the road curvature model of Fig. 6. The improvement by this development is clearly observable in the example, since using the newer approach the curved lane's and the straight lane's vehicle groups are appropriately separated.

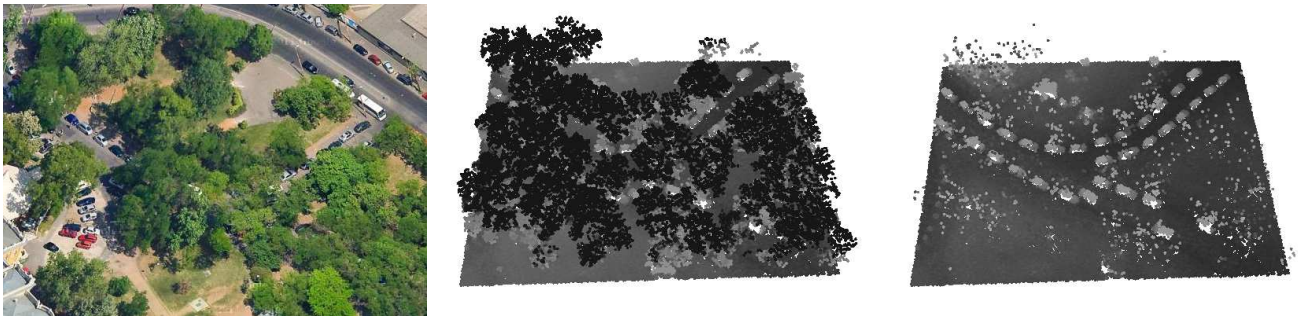
F. Relevance study of the different energy terms

The configuration energy of the L^{2}MPP model is composed by fusing various data terms and prior terms. For studying the relevance of the different features, we have tested the model with skipping selected components from the energy term, one after an other. Quantitative results regarding the test Data set #3 are listed in Table VI, and a few sample images are shown in Fig. 15. The first four rows of Table VI correspond to the data model verification. In the considered test sets, skipping the intensity feature f^{it} results only in a slight deficit of performance, which fact also confirms that the model is not very sensitive to the lack of calibrated intensity information. By ignoring also the vehicle evidence feature f^{ve} the number of false positive hits (FP) increases significantly, since the algorithm may detect false vehicles in *car sized holes* of the projected map, especially at the border of roof and terrain regions (Fig. 15(a)). Since vehicles are usually separated by background areas, without the internal background term f^{ib} some cars can be merged into the same object, or the detected shapes can significantly overhang the real car silhouettes (Fig. 15(b)). With skipping the external background feature, the object level performance does not decrease drastically, but the pixel level rates become lower since the car shapes are not completely recovered (Fig. 15(c)).

As for the prior energy terms, the $I(u, v)$ component has a crucial role to avoid multiple detections at a given vehicle position, therefore it cannot be skipped. As the fifth row of Table VI demonstrates, ignoring the *width uniformity* prior component results in slightly decreased pixel level recall and precision rates.

G. Dependence on point cloud resolution

To test the sensitivity of the proposed L^{2}MPP method w.r.t. point density reduction, we have downscaled the point clouds of Data sets $\#3$ and $\#6$. Comparative recognition rates obtained on data samples with 4 respectively 8 points/m^2 densities are displayed in Table. VII. In these regions the performance with $1/2$ density downsampling drops 4% at object and 3-7 % at pixel levels.



(a) Aerial photo (not used) (b) Input Lidar point cloud (c) Lidar data filtered by echo number

Fig. 14. Challenging data sample with a curved crossroad hidden by dense tree crowns (point intensity is related to elevation). A significant part of upper vegetation has been removed based on echo number, however the point density under the trees is usually less uniform than in clearly visible surfaces.

(a) Ignored f^{ve} (b) Ignored f^{ib} (c) Ignored f^{eb} (d) Complete L^2MPP (e) Ground truth

Fig. 15. Testing the significance of the individual energy terms on a sample part of Fig. 9. Ignoring the f^{ve} vehicle evidence features yields several false alarms, without the internal background f^{ib} term we get elongated shapes, without the external background f^{eb} shortened rectangles.

TABLE VII
SENSITIVITY OF THE PROPOSED L^2MPP ON REDUCED POINT CLOUD DENSITY.

Data set	Density	Obj. F-rate	Pix. F-rate
#3	4 pts/m^2	92	71
	8 pts/m^2	96	74
#6	4 pts/m^2	94	81
	8 pts/m^2	98	88

VI. CONCLUSIONS AND FUTURE WORK

This paper has proposed a novel Two-Level MPP model for joint extraction of vehicles and traffic segments in airborne laser point cloud data. The efficiency of the approach has been tested with real-world Lidar measurements, and its advantages versus four reference methods have been demonstrated. Although the vehicles are grouped based on similar orientation, with calculating a relative turning angle considering the road side contour, complex vehicle arrangement patterns could be recognized such as traveling cars in strongly curved exit ramps. For future work, the consideration of the 'shape shearing' effect [1] for vehicle motion estimation can be integrated into the Marked Point Process framework in a straightforward way by exchanging the five-parameter rectangle model to six-parameter parallelogram models.

REFERENCES

[1] W. Yao, S. Hinz, and U. Stilla, "Extraction and motion estimation of vehicles in single-pass airborne LiDAR data towards urban traffic analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, pp. 260–271, 2011.

[2] C. Benedek, T. Szirányi, Z. Kato, and J. Zerubia, "Detection of object motion regions in aerial image pairs with a multi-layer Markovian model," *IEEE Trans. Image Processing*, vol. 18, no. 10, pp. 2303–2315, 2009.

[3] L. Eikvil, L. Aurdal, and H. Koren, "Classification-based vehicle detection in high-resolution satellite images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 1, pp. 65 – 72, 2009.

[4] J. Leitloff, S. Hinz, and U. Stilla, "Vehicle detection in very high resolution satellite images of city areas," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 7, pp. 2795–2806, 2010.

[5] S. Hinz and U. Stilla, "Car detection in aerial thermal images by local and global evidence accumulation," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 308 – 315, 2006.

[6] M. Kirchhof and U. Stilla, "Detection of moving objects in airborne thermal videos," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 61, no. 3–4, pp. 187 – 196, 2006.

[7] W. Yao, S. Hinz, and U. Stilla, "Automatic analysis of traffic scenario from airborne thermal infrared video," in *XXI. ISPRS Congress*, vol. XXXVII-B3a of *ISPRS Archives Photogram. Rem. Sens. and Spat. Inf. Sci.*, pp. 223–228. Beijing, China, 2008.

[8] J.M. Munoz-Ferreras, F. Perez-Martinez, J. Calvo-Gallego, A. Asensio-Lopez, B. P. Dorta-Naranjo, and A. Blanco-del Campo, "Traffic surveillance system based on a high-resolution radar," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1624–1633, 2008.

[9] S. Suchandt, H. Runge, H. Breit, U. Steinbrecher, A. Kotenkov, and U. Balss, "Automatic extraction of traffic flows using TerraSAR-X along-track interferometry," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 48, no. 2, pp. 807–819, 2010.

[10] S.V. Baumgartner and G. Krieger, "Fast GMTI algorithm for traffic monitoring based on a priori knowledge," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4626–4641, 2012.

[11] O. Maksymiuk, M. Schmitt, A.R. Brenner, and U. Stilla, "First investigations on detection of stationary vehicles in airborne decimeter resolution sar data by supervised learning," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Munich,Germany, 2012, pp. 3584–3587.

[12] U. Stilla, E. Michaelsen, U. Soergel, S. Hinz, and H.J. Ender, "Airborne monitoring of vehicle activity in urban areas," in *XX. ISPRS Congress*, vol. XXXV-B3 of *ISPRS Archives Photogram. Rem. Sens. and Spat. Inf. Sci.*, pp. 973–979. Istanbul, Turkey, 2004.

[13] V. Ussyshkin and L. Theriault, "Advances in discrete return technology for 3D vegetation mapping," *Remote Sensing*, vol. 3, no. 3, pp. 416–434, 2011.

[14] W. Wagner, "Radiometric calibration of small-footprint airborne laser scanner measurements: Basic physical concepts," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 10, pp. 505–513, 2010.

[15] W. Yao and U. Stilla, "Comparison of two methods for vehicle extraction from airborne LiDAR data toward motion analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 607–611, 2011.

[16] Á. Rakusz, T. Lovas, and Á. Barsi, "Lidar-based vehicle segmentation," in *XX. ISPRS Congress*, vol. XXXV-2 of *ISPRS Archives Photogram. Rem. Sens. and Spat. Inf. Sci.*, pp. 156–159. Istanbul, Turkey, 2004.

[17] B. Yang, P. Sharma, and R. Nevatia, "Vehicle detection from low quality aerial LIDAR data," in *IEEE Workshop on Applications of Computer Vision (WACV)*, 2011, pp. 541 –548.

[18] W. Yao, S. Hinz, and U. Stilla, "Automatic vehicle extraction from airborne LiDAR data of urban areas aided by geodesic morphology," *Pattern Recogn. Letters*, vol. 31, no. 10, pp. 1100 – 1108, 2010.

[19] C.K. Toth and D. Grejner-Brzezinska, "Extracting dynamic spatial data from airborne imaging sensors to support traffic flow estimation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 61, no. 3-4, pp. 137 – 148, 2006.

[20] C. Wang and N.F. Glenn, "Integrating LiDAR intensity and elevation data for terrain characterization in a forested area," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 3, pp. 463–466, 2009.

[21] F. Samadzadegan, M. Hahn, and B. Bigdeli, "Automatic road extraction from LIDAR data based on classifier fusion," in *Joint Urban Remote Sensing Event*, Shangai, China, 2009, pp. 1–6.

[22] W. Yao, S. Hinz, and U. Stilla, "Vehicle activity indication from airborne LiDAR data of urban areas by binary shape classification of point sets," in *ISPRS Workshop Object Extraction for 3D City Models, Roads and Traffic (CMRT)*, Paris, France, 2009, pp. 187–192.

[23] X. Descombes, R. Minlos, and E. Zhizhina, "Object extraction using a stochastic birth-and-death dynamics in continuum," *Journal of Mathematical Imaging and Vision*, vol. 33, pp. 347–359, 2009.

[24] A. Börcs and C. Benedek, "Urban traffic monitoring from aerial LIDAR data with a two-level marked point process model," in *International Conference on Pattern Recognition (ICPR)*, Tsukuba City, Japan, 2012, pp. 1379–1382.

[25] J. Zhao and S. You, "Road network extraction from airborne LiDAR data using scene context," in *International Workshop on Point Cloud Processing*, Providence, USA, 2012, pp. 9–16.

[26] A. Boyko and T. Funkhouser, "Extracting roads from dense point clouds in large scale urban environment," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 6, pp. 2–12, 2011.

[27] D. Chai, W. Forstner, and F. Lafarge, "Recovering Line-networks in Images by Junction-Point Processes," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, 2013.

[28] C. Benedek and M. Martorella, "Moving target analysis in isar image sequences with a multiframe marked point process model," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2234–2246, 2014.

[29] P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition, 2003.

[30] F. Chatelain, X. Descombes, F. Lafarge, C. Lantuejoul, C. Mallet, R. Minlos, M. Schmitt, M. Sigelle, R. Stoica, and E. Zhizhina, *Stochastic geometry for image analysis*, Digital Signal and Image Processing. Wiley-ISTE, 2011.

[31] Y. Verdié and F. Lafarge, "Detecting parametric objects in large scenes by Monte Carlo sampling," *International Journal of Computer Vision*, vol. 106, no. 1, pp. 57–75, 2014.

[32] C. Benedek, O. Krammer, M. Janóczki, and L. Jakab, "Solder paste scooping detection by multi-level visual inspection of printed circuit boards," *IEEE Trans. on Industrial Electronics*, vol. 60, no. 6, 2013.

[33] A. Gamal Eldin, X. Descombes, G. Charpiat, and J. Zerubia, "Multiple Birth and Cut Algorithm for Multiple Object Detection," *Journal of Multimedia Processing and Technologies*, vol. 1, no. 4, pp. 260–276, 2010.

[34] A. Börcs and C. Benedek, "A marked point process model for vehicle detection in aerial LIDAR point clouds," in *XXII. ISPRS Congress*, vol. I-3 of *ISPRS Annals Photogram. Rem. Sens. and Spat. Inf. Sci.*, pp. 93–98. Melbourne, Australia, 2012.

[35] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *International Journal of Computer Vision*, vol. 99, no. 1, pp. 69–85, 2012.

[36] C. Benedek, D. Molnár, and T. Szirányi, "A dynamic MRF model for foreground detection on range data sequences of rotating multi-beam Lidar," in *International Workshop on Depth Image Analysis (WDIA)*, vol. 7854 of *Lecture Notes in Computer Science*, pp. 87–96. Tsukuba City, Japan, 2012.

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

[38] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, 2004.

[39] F. Chatelain, X. Descombes, and J. Zerubia, "Parameter estimation for marked point processes. application to object extraction from remote sensing images," in *Energy Minimization Methods in Comp. Vision and Pattern Recogn.*, vol. 5681 of *Lecture Notes in Computer Science*, pp. 221–234. Bonn, Germany, 2009.

[40] H.W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, pp. 83–97, 1955.

[41] F. Mokhtarian and S. Abbasi, "Shape similarity retrieval under affine transforms," *Pattern Recognition*, vol. 35, no. 1, pp. 31–41, 2002.

Attila Börcs received the M.Sc. degree in computer sciences in 2012 from the Pázmány Péter Catholic University, Budapest. He is currently pursuing the Ph.D. degree at the Department of Control Engineering and Information Technology (IIT) of the Budapest University of Technology and Economics (BME). He is also a member of the Distributed Events Analysis Research Laboratory, at the Institute for Computer Science and Control, Hungarian Academy of Sciences. His research interests include aerial and terrestrial Laser scanning, and object recognition in point clouds.

Csaba Benedek received the M.Sc. degree in computer sciences in 2004 from the Budapest University of Technology and Economics (BME), and the Ph.D. degree in image processing in 2008 from the Pázmány Péter Catholic University, Budapest. Starting from October 2008, he worked for 12 months as a postdoctoral researcher with the Ariana Project Team at INRIA Sophia-Antipolis, France. He is currently a senior research fellow with the Distributed Events Analysis Research Laboratory, at the Institute for Computer Science and Control of the Hungarian Academy of Sciences. He has been the national project leader of the Array Passive ISAR adaptive processing (APIS) project funded by the European Defense Agency. Currently he is leading the DUSIREF remote sensing project of the European Space Agency. His research interests include Bayesian image segmentation and object extraction, change detection, scene recognition and reconstruction from Lidar point clouds and remotely sensed data analysis.