HAL
open science

Ground Estimation and Point Cloud Segmentation using SpatioTemporal Conditional Random Field

Lukas Rummelhard, Anshul Paigwar, Amaury Nègre, Christian Laugier

To cite this version:

Lukas Rummelhard, Anshul Paigwar, Amaury Nègre, Christian Laugier. Ground Estimation and Point Cloud Segmentation using SpatioTemporal Conditional Random Field. IEEE Intelligent Vehicles Symposium (IV), Jun 2017, Redondo Beach, United States. pp.1105 - 1110, 10.1109/IVS.2017.7995861 . hal-01579095

HAL Id: hal-01579095

https://hal.inria.fr/hal-01579095

Submitted on 30 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ground Estimation and Point Cloud Segmentation using SpatioTemporal Conditional Random Field

Lukas Rummelhard*†, Anshul Paigwar*, Amaury Nègre*‡ and Christian Laugier*
*Inria, Chroma, name.surname@inria.fr
†currently CEA, name.surname@cea.fr
‡currently Univ. Grenoble Alpes, CNRS, GIPSA-lab, name.surname@gipsa-lab.fr

***Abstract*—Whether it be to feed data for an object detection-and-tracking system or to generate proper occupancy grids, 3D point cloud extraction of the ground and data classification are critical processing tasks, on their efficiency can drastically depend the whole perception chain. Flat-ground assumption or form recognition in point clouds can either lead to systematic error, or massive calculations. This paper describes an adaptive method for ground labeling in 3D Point clouds, based on a local ground elevation estimation. The system proposes to model the ground as a Spatio-Temporal Conditional Random Field (STCRF). Spatial and temporal dependencies within the segmentation process are unified by a dynamic probabilistic framework based on the conditional random field (CRF). Ground elevation parameters are estimated in parallel in each node, using an interconnected Expectation Maximization (EM) algorithm variant. The approach, designed to target high-speed vehicle constraints and performs efficiently with highly-dense (Velodyne-64) and sparser (Ibeo-Lux) 3D point clouds, has been implemented and deployed on experimental vehicle and platforms, and are currently tested on embedded systems (Nvidia Jetson TX1, TK1). The experiments on real road data, in various situations (city, countryside, mountain roads,...), show promising results.**

I. Introduction and related work

Fully autonomous driving is an important but challenging goal, for which a reliable perception of the local environment is crucial[1]. Accurate detection of the navigable space and classification of obstacles are key tasks for intelligent vehicles, to be able to estimate the risks on its path and the surrounding motion. Many sensor outputs consist in raw impact point clouds, from which must be differentiated the points related to actual obstacles and the ones related to clear areas. Laser range scanners (3D-LiDARs), for example, generate high resolution 3D point clouds of the environment while remaining unaffected by varying illumination, making them popular and widely used in robotics. Because of their relative cost in comparison to other systems (camera, radar, ultrasonic,...), which is still prohibitive for many constructors, LiDAR sensors have not yet reached consumer market vehicle constraints, but recent key technological progress (Solid State LiDARs) should make them more affordable.

From this raw sensor data, most perception approaches rely, at some point, on a classification between obstacle data, relevant for occupancy grid generation or object detection-and-tracking, and the data relative to the ground. If a simple height thresholding can be used in very simple situations, a much more sophisticated approach needs to be adopted in most cases. Many methods rely on previously-generated maps, and concentrate on precise localization in those maps, thus extracting the local ground profile, but such maps, although being impressively developing in terms of accuracy and coverage, are not always available, nor really necessary in local perception. Other approaches focus on pure data point analysis, and then have to deal with undefined ground form, and often sparse data, in particular in high-speed vehicle perception. In many cases, as the one presented in this paper, those methods are designed to be embedded with real-time performances on computationally and energetically limited systems, shared by a full perception chain, which can include important computing-power consuming components (systems of advanced situation awareness, motion planning, etc.). Computational efficiency is then a critical aspect.

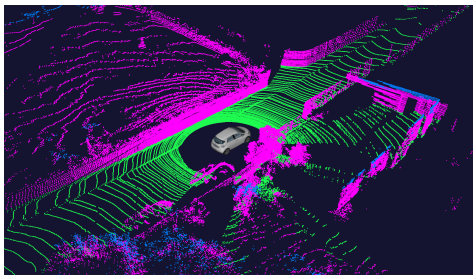

Fig. 1. Illustration of 3D Point cloud segmentation following the road slope. Ground points are green, obstacles are pink.

Various segmentation methods of 3D LiDAR point clouds are compared in [2]. Zhang et al. in [3] describe existing methods and classify them into four categories, namely :

- *Elevation map methods:* Used by many teams in DARPA Urban Challenge [4], 3D points are projected as 2.5D grid and a Min-Max elevation map is used for segmentation. These methods are efficient but suffer from under-segmentation and also errors in case of overhanging structures (like treetops and bridges)
- *Ground modeling methods:* Himmelsbach et. al [5] proposed a 2D-line-extraction-based fast algorithm to estimate local ground plane but the method is not suitable

for curved terrains. This problem was addressed by Chen et. al in [6] using an approach based on Gaussian process regression in the polar grid map for the ground segmentation. However, the space is divided on independent angular sectors so that the overall ground elevation continuity is not assured.

- *Methods based on the relationship between adjacent points:* Moosmann et al. in [7] presented Local Convexity criteria that uses local geometric features for segmentation. This approach keeps the full 3D information delivered by the sensor unlike many other popular approaches.
- *MRF - based methods:* Several MRF-based road detection methods in [8], [9] have the potential for ground segmentation. These methods use the gradient cues of the road geometry to construct MRF and implement a belief propagation (BP) algorithm to classify the surrounding into different regions. Zhang et al. extended from them in [3] and clearly demonstrated the ability of MRF-based approaches to accurately segment point clouds even in undulated roads such as down/uphill. But use of belief propagation for inference on MRF is computationally expensive, which limits real-time performance perspectives.

Conditional Random Fields (CRF) [10], [11] are a variant of MRFs, and widely used for segmentation and recognition problems in computer vision and machine learning. In comparison to generative models, including MRF and HMM, CRF are rather discriminative in nature and allow arbitrary relationship among data. Observations may have complex dependencies and modeling them can lead to intractable models, while ignoring them can increase performance. CRF models the conditional distribution p(y|x) over latent variables given observations directly, which is all that is needed for segmentation.

Lu et. al. in [12] have proposed a new "hybrid" CRF model to automatically extract Digital Terrain Model(DTM) from Digital Surface Model(DSM) generated from airborne LiDARs. Their model contains discrete random variables which represent whether a specific point in a point cloud belongs to the ground or not. Also, their model contains continuous hidden random variables which represent the height of the underlying ground surface at that point.

Wang and Ji in [13] have proposed a probabilistic framework based on a Dynamic Conditional Random Field (DCRF) model to integrate temporal and spatial constraints for object segmentation in image sequences. A similar spatiotemporal CRF model for video segmentation was presented in [14], probabilistic inference for obtaining optimal labeling of pixels as foreground or background was done by Sum-product loopy belief propagation.

In this paper will be presented an approach based on a SpatioTemporal Conditional Random Field (STCRF) for ground segmentation, which was inspired by the work of Lu et. al. in [12], and the works in [3], [8], [9]. To address the problem of insufficient measurement points, a dynamic Bayesian framework is used, to incorporate temporal constraints among successive segmentation fields as in [13], [14], since spatial information alone may not be sufficient to accurately model the ground profile. For the inference on STCRF, the Maximum A Posteriori (MAP) estimate is found using a variant of the Expectation Maximization (EM) algorithm[15] which computational efficiency allows real-time performances. The method was experimented on real data acquired by different types of LIDARs, Velodyne HDL-64 (dense data) and 4 Ibeo-Lux 4-laser (sparser data), equipped on our experimental Renault Zoe. Experimental results presented promising performances in uneven terrain, particularly in comparison to other local ground segmentation methods.

II. Proposed Model

A. Overview

The ground surface is modeled as CRF and represented by a regular 2D lattice on XY plane. To each node $N_i = (nx_i, ny_i)$ of the lattice is associated a continuous hidden random variable $G_i = (h_i, sx_i, sy_i)^T \in \mathbb{R}^3$ representing the estimated elevation and two directional slopes of ground. Let $G = (G_i, ..., G_n)$ be the set of all such variables in the lattice, where n is the number of nodes. The LiDAR measurement consists of a set of 3D locations $(x_j, y_j, z_j) \in \mathbb{R}^3$. Each measurement is associated with its closest node. The set of observation indexes associated with the node N_i is noted $\mathcal{M}_i$. For every point j is defined a binary random variable $C_j \in \{0, 1\}$ indicating the point classification, i.e., whether the point belongs to the ground (0) or not (1). Let Z denotes the set of all the LiDAR measurements and C the set of all binary random variables associated to corresponding measurements. The set of indexes corresponding to the neighborhood of a node N_i is noted $\mathcal{N}_i$. G_i^{t-1} represents the random variable associated with node N_i at previous time step.

The goal of the processing is to estimate the unknown variables G and C, given the observed variables Z and the random distribution of variables G at previous time step G_i^{t-1}.

B. Ground Model

To model the interaction between the nodes, a Conditional Random Field (CRF) approximation is used. A variable G_i is considered to only depend on its spatial neighborhood $\{G_k | k \in \mathcal{N}_i\}$ (spatial clique), temporal neighborhood G_i^{t-1}(temporal clique) and on observations and hidden variables $\{Z_j, C_j | j \in \mathcal{M}_i\}$ (data clique). A graphical representation of the STCRF is shown on figure 2. These clique potentials in Gibbs distribution form are defined as follows :

- The measurement potential:

$$\psi = \frac{1}{W_1} exp(-\sum_i \alpha \sum_{j \in \mathcal{M}_i} c_j ||z_j - H_{ij} G_j||^2) \qquad (1)$$

where W_1 is a partition function, α a scalar parameter expressing the confidence in the measurements and H_{ij} a measurement matrix depending on the relative horizontal positions of the node and the measurement.

$$W_1 = \sum_G exp(-\sum_i \alpha \sum_{j \in \mathcal{M}_i} c_j ||z_j - H_{ij} G_j||^2)$$

$$H_{ij} = [\ 1 \quad x_i - nx_i \quad y_i - ny_i\ ]$$

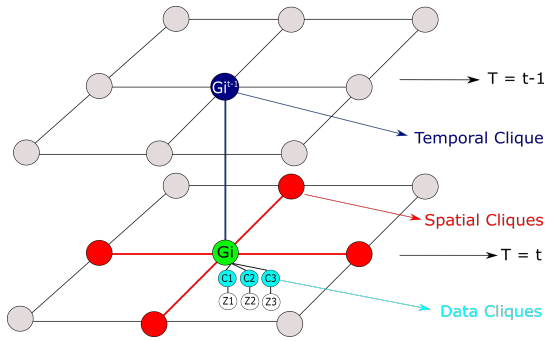

Fig. 2. SpatioTemporal Conditional Random Field (STCRF) model.

- Node smoothness potential:

$$\Phi = \frac{1}{W_2} exp(-\sum_i \beta \sum_{j \in \mathcal{N}_i} ||G_i - F_{ij} G_j||^2) \tag{2}$$

where W_2 is a partition function, β a scalar parameter representing the confidence in the elevation estimates of the neighboring cells and F_{ij} a matrix depending on the relative positions of two nodes:

$$W_2 = \sum_G exp(-\sum_i \beta \sum_{j \in \mathcal{N}_i} ||G_i - F_{ij} G_j||^2)$$

$$F_{ij} = \begin{bmatrix} 1 & nx_j - nx_i & ny_j - ny_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Temporal interaction potential:

$$\xi = \frac{1}{W_3} exp(-\sum_i \gamma ||G_i - Q_i G_i^{t-1}||^2) \tag{3}$$

where W_3 is a partition function, γ a scalar parameter representing the confidence in the elevation estimates of the cells at previous time step and Q_i a transition matrix that transforms G_i^{t-1}, representing the value of node at previous time step, to the state of node G_i at current time step. In this paper an Identity matrix will be used.

$$W_3 = \sum_G exp(-\sum_i \gamma ||G_i - Q_i G_i^{t-1}||^2)$$

C. Vehicle motion model

To find the estimate of a ground node at previous time step G_i^{t-1}, the previous grid is transformed into the current reference frame, using the transform computed from the vehicle displacement and orientation change, calculated from the fusion of IMU, GPS and odometry data. For each node G_i the value at previous time is then interpolated. In the areas corresponding to newly-discovered regions, the values at previous time step are set undefined, while taken out areas can simply be forgotten, or stored in a long-term map.

III. Inference

A. Gaussian Model and iterative Expectation Maximization method

The elevation estimation of each cell variable $\{G_i\}_{i=1}^n$ is computed using Maximum a Posterior inference. The best solution $\{\tilde{g}_i\}$ for the ground estimation is the value that maximize the posterior probability $\{\tilde{g}_i\} = \text{argmax}_{\{g_i\}}(P(G = \{g_i\}|C, Z, G^{t-1}, \theta))$

The computation of the exact solution is intractable, since it would imply to sum over all possible values (in continuous space) and over all possible cliques of the graph.

In order to simplify the inference, a Gaussian model is chosen for the ground state. The distribution over each node state G_i is represented by a mean vector $\tilde{G}_i$ and a covariance matrix M_i. In the following, to simplify the formula, the information vector X_i and information matrices P_i will be used, defined as follows:

$$P_i = M_i^{-1}$$

$$X_i = M_i^{-1} \tilde{G}_i$$

Then, to approximate the full distribution over G_i and C_i, an iterative Expectation-Maximization (EM) [15] algorithm is used. The Expectation (E) step estimates the probability distribution over the point classification C_i while the Maximization (M) step uses this distribution to estimate the ground state distribution. Those 2 steps are alternatively repeated in a loop over a number of iterations.

B. EM initialization

At the beginning, and on the border of the grid when the vehicle moves, the initial state of the ground is unknown. To initialize the EM process, the ground mean state is set to 0 height, with a null slope. The covariance matrix is initialized with great diagonal coefficients, so that these initialization values have few impact on the final result.

C. E-step

The E-step computes the probability for each measurement z_j to be a ground point. This probability $c_i = p(C_j = 1)$ is estimated considering the approximated state G_i of the closest node *i* (estimated on the M-step of the previous iteration).

This probability is computed using the relative height dz_j between the local ground surface and the measurement (see figure 3) as follows:

$$dz_j = z_i - H_{ij} X_i$$

$$c_j = \begin{cases} exp(-\frac{dz^2}{2.\sigma_{up}^2}) & if\ dz \geq 0 \\ exp(-\frac{dz^2}{2.\sigma_{down}^2}) & if\ dz < 0 \end{cases}$$

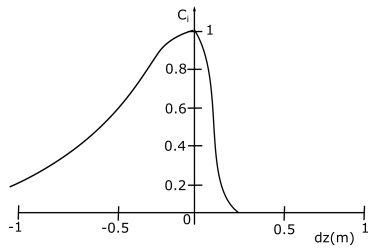

Fig. 3. Likelihood function used in the E-step to calculate the probability for a measurement to belong to the ground. It only depends on the vertical distance dz between the LiDAR point and the ground node. The asymmetry with respect to 0 allows to give more importance to lower points and to converge to the bottom of obstacles.

D. M-step

The M-step estimates the ground state distribution for each node given the measurement values, the classification and the neighborhood interactions. To approximate the distribution, the information vector X_i and information matrix P_i are updated for each node i as follows:

$$X_i^{k+1} = \alpha \sum_{j \in \mathcal{M}_i} c_j z_j H_{ij}^T + \beta \sum_{j \in \mathcal{N}_i} F_{ij}^T X_j^k + \gamma Q_i^T X_i^{t-1}$$

$$P_i^{k+1} = \alpha \sum_{j \in \mathcal{M}_i} c_j H_{ij}^T H_{ij} + \beta \sum_{j \in \mathcal{N}_i} F_{ij}^T P_j^k F_{ij} + \gamma Q_i^T P_i^{t-1} Q_i$$

IV. Experimental Results

A. Experimental Platform

For the experiments, a Renault Zoe car (figure 4) has been equipped with a Velodyne HDL64 on the top, covering a 360-degree field of view (FOV), 3 Ibeo Lux LiDARs on the front covering 160-degree FOV and one on the back covering 85-degree FOV. Xsens GPS and IMU provide vehicle velocity and orientation. Two IDS cameras, on the front and on the back, are also mounted.

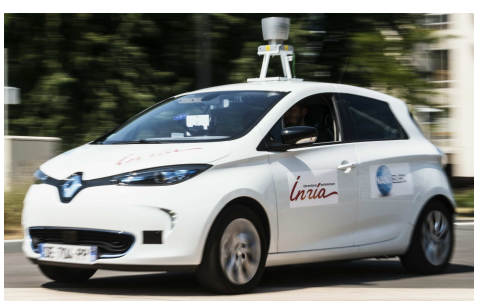

Fig. 4. Experimental Platform: Renault Zoe car equipped with Velodyne HDL64, 4 Ibeo Lux LiDARs, Xsens GPS and IMU and cameras.

B. Runtime

The presented algorithm has been implemented in C++, accelerated using the Nvidia Cuda parallel architecture in order to run on a Graphic Processing Unit (GPU). Robot Operating System (ROS) is used to ease compatibility and the connection of specific sensor drivers and the various software components. The testing specification (modest hardware) is shown in table I.

TABLE I

CPU	Intel Xeon(R) W3520 @ 2.6GHz
Memory (RAM)	8.0 GB
GPU	Quadro 2000 - 2GB
Programming Language	C++ and CUDA
Grid map size	120m x 80m (1m x 1m nodes)
Number of EM iterations	10

Computational performances of the algorithm with different LiDAR sensors are as shown in table II.

TABLE II

Sensors	Data packet rate	Avg. data points per packet	Avg. rate of performance
Velodyne HDL-64E	9.8 Hz	100,000	6.8 Hz
4 x Ibeo Lux 4 layers	24.5 Hz	2,550	21 Hz

Those numbers are found using a middle-performance set up. On the experimental platforms, equipped with Titan X, Tegra X1 and K1, the proposed approach will largely satisfy real-time application constraints, as first tests confirm. The algorithm performance was further improved by iterating between E-step and M-step directly in the same kernel, and using Inter-Block GPU Communication via Fast Barrier Synchronization as explained in [16] after every EM step.

C. Qualitative Analysis

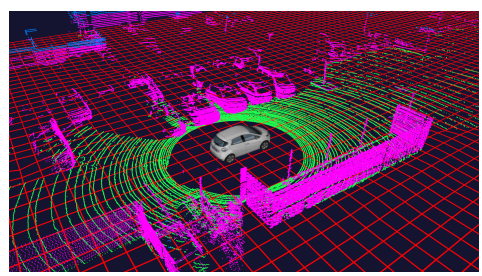

Fig. 5. Results in a parking lot with vehicles.

The proposed method has been evaluated on many real road data scenarios, the data being acquired while manually driving our equipped car in inner city traffic, countryside scenes, on highways, mountain roads. As no precise ground truth data is available, the performance evaluation mainly consists in qualitative analyses. In the following the parameters are fixed to $\alpha = 1$,$\beta = 0.5$,$\gamma = 0.2$, $\sigma_{up} = 0.05m$, $\sigma_{down} = 0.5m$.

On the illustrations, the point clouds are segmented into ground-related points in green, and obstacle-related points in pink. On most of them is also represented a red mesh, partly representing the ground elevation model : to every point of the mesh corresponds the average estimated height of the ground in the node at this location. It is worth noting that

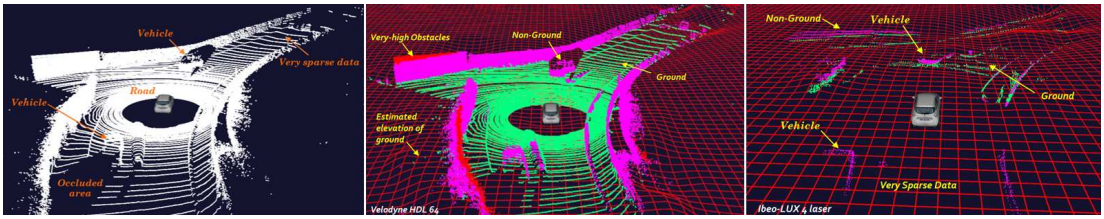

Fig. 6. (a) Typical 3D point cloud generated by Velodyne LiDAR, (b) Point cloud segmentation between ground (green points) and non-ground (pink points), and estimated average elevation of the terrain (red grid) (c) Point Cloud Segmentation on 4-Ibeo Lux LiDAR data and estimated elevation of terrain.

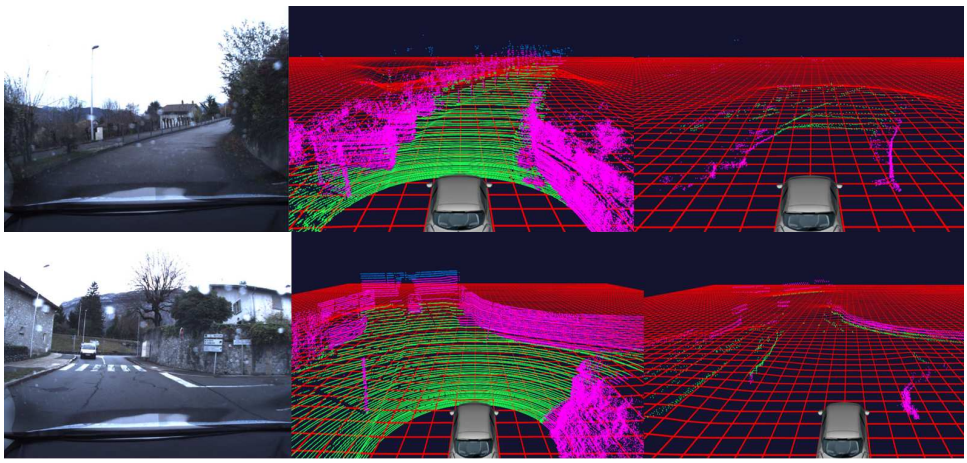

Fig. 7. Ground segmentation of 3D point clouds from Velodyne and 4-Ibeo Lux LiDAR, and estimated ground elevations, on mountain road and rural city.

in long-term dataless areas, while the average height tends towards 0, the variance tends towards infinity, so that the systems knows those values are meaningless. On figure 1 is depicted the simplest example of ground segmentation, as the ground is flat and obstacles are high, with sharp edges with the ground. On figure 5, the scene is a bit more complex, as some low-height obstacles and many with less-sharp edges with the ground could have deteriorated the estimation. The figures 6 and 7 show results on rural mountain roads, with important slope variations, with different sensor inputs. If Velodyne point segmentation confirms the method efficiency in these difficult environment, the most interesting results are to be seen with the IBEO data : with few data inputs, the system is able to correctly assess navigable space and real obstacles, thanks to the spatiotemporal conditioning. Figure 8 shows the concrete effect of temporal filtering on the model, especially the conservation of the estimated ground profile in case of momentary obstacle interposition. Figure 9 shows latest results in city environments, using the KITTI dataset. Not only those results show persuasive outputs, but the use of these partially labeled data (in terms of ground/obstacle segmentation and ground elevation), widely used in the domain, will lead to more quantitative evaluations and comparisons.

V. Conclusion and Future work

In this paper was presented a 3D point cloud ground segmentation system, based on a dynamic estimation of local ground elevation and slope. The system models the ground as a Spatio-Temporal Conditional Random Field, dividing the surrounding into interconnected elevation cells, affected by local observations and spatio-temporal dependencies. Ground elevation parameters are estimated in parallel in each cell, using an interconnected Expectation Maximization algorithm variant. The computations are accelerated using GPU technologies, allowing real-time performances on embedded devices and experimental platforms. Experiments on various environments (city, countryside, mountain roads,...) using different sensors (Velodyne-64, Ibeo-Lux) show favorable results. If qualitative analysis of these results are promising, a quantitative analysis and rigorous comparison to other methods are to be defined and performed. Future works will consist in those, integration of dynamic knowledge on moving objects in the model and automatic generation of occupancy grids using those results, defining a smart sensor model.

Acknowledgment

This work has been developed within the scope of ”Institut de Recherche Technologique” NanoElec, founded by the French program ”Investissements d’Avenir” ANR-10-AIRT-05.

References

[1] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, *et al.*, “A perception-driven autonomous urban vehicle,” *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

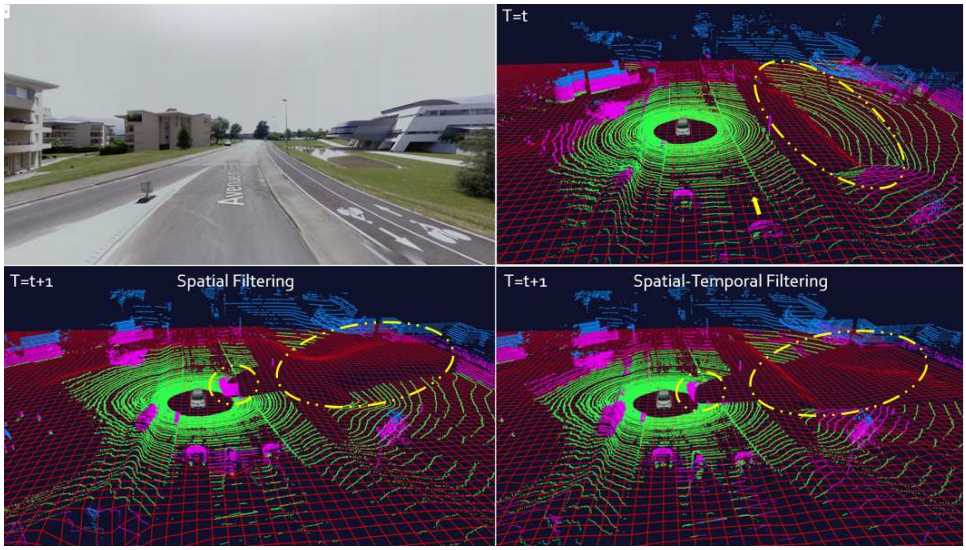

Fig. 8. Scene where the road is higher than the ground on the right, in front of the Inria building. Top right is the result at a given time t, accurately describing the form of the ground. At the bottom are depicted the ground estimation at the next time step t+1, without temporal conditioning and with. In yellow-circled areas can be observed the ground figure conservation in non-visible regions (obstructed by another car) when temporal constraints are added.

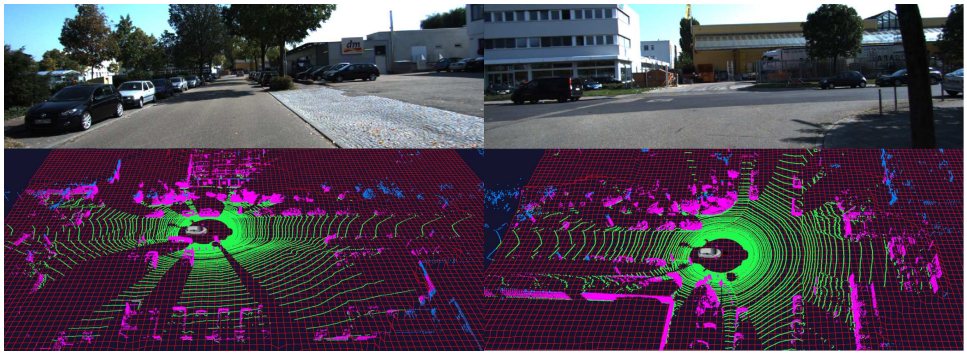

Fig. 9. Results of our approach on KITTI Raw dataset in an urban environment with vehicle traffic.

[2] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3d lidar point clouds," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 2798–2805, May 2011.

[3] M. Zhang, D. D. Morris, and R. Fu, "Ground segmentation based on loopy belief propagation for sparse 3d point clouds," in *3D Vision (3DV), 2015 International Conference on*, pp. 615–622, Oct 2015.

[4] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[5] M. Himmelsbach, F. v. Hundelshausen, and H. J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 560–565, June 2010.

[6] T. Chen, B. Dai, R. Wang, and D. Liu, "Gaussian-process-based real-time ground segmentation for autonomous land vehicles," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 3, pp. 563–582, 2014.

[7] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *Intelligent Vehicles Symposium, 2009 IEEE*, pp. 215–220, June 2009.

[8] C. Guo, W. Sato, L. Han, S. Mita, and D. McAllester, "Graph-based 2d road representation of 3d point clouds for intelligent vehicles," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pp. 715–721, June 2011.

[9] J. Byun, K.-i. Na, B.-s. Seo, and M. Roh, *Drivable Road Detection with 3D Point Clouds Based on the MRF for Intelligent Vehicle*, pp. 49–60. Cham: Springer International Publishing, 2015.

[10] S. Prince, *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.

[11] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and TrendsÂ® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[12] W. L. Lu, K. P. Murphy, J. J. Little, A. Sheffer, and H. Fu, "A hybrid conditional random field for estimating the underlying ground surface from airborne lidar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 2913–2922, Aug 2009.

[13] Y. Wang and Q. Ji, "A dynamic conditional random field model for object segmentation in image sequences," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 264–270 vol. 1, June 2005.

[14] L. Zhang and Q. Ji, "Segmentation of video sequences using spatial-temporal conditional random fields," in *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*, pp. 1–7, IEEE, 2008.

[15] M. Tanner, *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*. Springer Series in Statistics, Springer New York, 1997.

[16] S. Xiao and W. c. Feng, "Inter-block gpu communication via fast barrier synchronization," in *Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, pp. 1–12, April 2010.