



HAL
open science

Towards a Reference Data Generation Framework for Performance Assessment of Perception Systems

Rémi Defraiteur, Sergio Alberto Rodriguez Florez, Marie-Anne Mittet, Roger Reynaud, Nicole El Zoghby

► **To cite this version:**

Rémi Defraiteur, Sergio Alberto Rodriguez Florez, Marie-Anne Mittet, Roger Reynaud, Nicole El Zoghby. Towards a Reference Data Generation Framework for Performance Assessment of Perception Systems. IEEE Intelligent Transportation Systems Conference (ITSC), Oct 2019, Auckland, New Zealand. 10.1109/ITSC.2019.8917536 . hal-02393078

HAL Id: hal-02393078

<https://hal.science/hal-02393078>

Submitted on 4 Dec 2019

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.

Towards a Reference Data Generation Framework for Performance Assessment of Perception Systems

Rémi Defraiteur^{*†‡}, Sergio A. Rodríguez F.^{*†}, Marie-Anne Mittet[‡], Roger Reynaud^{*†} and Nicole El Zoghby[‡]

^{*}SATIE Laboratory CNRS Joint research unit - UMR 8029 - firstname.lastname@u-psud.fr

[†]ENS Paris-Saclay. Paris-Sud University. Paris-Saclay University, 91405, Orsay, France

[‡]Renault S.A.S - Guyancourt, France - (marie-anne.n.mittet, nicole.el-zoghby)@renault.com

Abstract—Sensors and their associated data fusion techniques, play a crucial role in Autonomous Vehicle (AV) decision-making applications. Accurately evaluate performance and reliability of the perception sources is an important task to be able to know the consistency of this data fusion. In this paper, a reference data generation framework for assessing perception sensors performances is proposed. Our approach relies on the complementary use of three data sources: a highly precise 3D map with semantic information, a High Density range finder sensor and a GNSS-RTK/INS localization unit. 3D map provides semantic knowledge of the environment and HD range finder precisely senses ego-vehicle surroundings. Finally, 3D map and HD scans are geometrically associated using positioning information in order to combine them and to infer reference data. Thorough experiments were conducted to evaluate and validate the proposed approach. As a proof of concept, performances of a LiDAR-based road plane detection method were evaluated, quantified and reported in terms of precision and recall.

I. INTRODUCTION

Majority of ADAS (Advanced Driver Assistance Systems) functionalities rely on data transmitted by integrated perception sensors to control the vehicle. Precise performance evaluation of embedded perception solutions is then an important task for AV and ADAS. It is therefore crucial to be able to evaluate performances and reliability of embedded perception solutions. In the aim of evaluating and validating perception solutions, existing approaches that have been proposed in the literature can be structured in three categories: manual, semi-supervised and automatic reference data creation.

Manual labeling: First and less complex method consists in generating referenced datasets by human manual processing. LiDAR (Light Detection And Ranging) datasets close to ADAS functions conditions of use correspond to sequences of point clouds measured from moving sensor. Sequences are recorded to be reusable. Each cloud of the whole sequences is manually segmented and object-level labeling is carried out by hired annotators or crowd-sourcing means. This is the case of KITTI Dataset [1], where the use of both hired annotators and crowd-sourcing were exploited to assign label objects to 3D bounding boxes manually defined. Some approaches were developed to simplify object labeling. In [2], labeling task is facilitated by optimizing points clouds using voxels which are

further hand-labeled. Similar techniques are used for images as in [3] where images are hand-labeled. In [4], a database composed of LiDAR and images is addressed. Point clouds are manually labeled and exploited further for referencing images. Finally, in [5], [6] software tools are created to facilitate and speed-up manual generation of reference data for 2D/3D object detection. The main advantage of manual labeling is the high accuracy of reference data created. However, manual labeling is time- and cost- consuming and is not a scalable solution. Indeed, hand-labeled data is specific to each dataset and cannot be applied on new datasets.

Semi-supervised approaches: In [7] a semi-supervised process for reference data generation method was developed in order to annotate scenes collected from KITTI optical flow dataset. This method uses the labeled 3D bounding boxes of KITTI to project them on 2D images and label the pixels of optical flow images. In [8], [9] stereo images are employed to reconstruct 3D point clouds. The created clouds are then classified by trained machine learning algorithm (e.g. random forest algorithm). Finally, the labeled data are re-projected in the 2D image plane. The disadvantage of these solutions is the changing of dimensions between 2D images and 3D point clouds that can bring errors in the data. Moreover, with machine learning algorithm trained on a single dataset, the risk of over-fitting can also bring classification errors in assigned labels.

Automatic methods: Finally, as far as we know, unsupervised approaches generating reference data are based on the use of synthetic data. Sensor simulation tools like Blensor [10], Helios [11] or Carla [12] are used to generate reference data. This is the case in [13], [14], [15] where labeled reference images are obtained in simulated environments. This solution allows the user to generate a big amount of reference data, sometimes very close to real data [16], [17], [18] and scenarios are easy to reproduce with new sensor solutions. However, since the environment is perfectly controlled, synthetic data are not impacted by external errors or other phenomena that only appear under real sensing conditions. To be closer to non-simulated data, reference data generated must take into considerations all the effects that could appear under real conditions.

To generate reference data needed for perception algorithms evaluation, our approach aims to generate reference data in a semi-supervised way and combines some advantages of the three cited approaches. It is done by augmenting perception data recorded with a simulated controlled environment. The main strength is the reproducibility and compatibility of our methodology with new datasets recorded, without needing manual labeling for each of the new datasets.

This paper is structured as follows : Section II describes the proposed semi-supervised approach for reference data generation allowing performance evaluation on perception algorithms. This strategy also provides a common referencing support to quantify consistent uncertainties of the perception functions as stated in section IV. Experiments are presented to assess our methodology. An implementation is also deployed to evaluate a 3D road detection algorithm performances. Finally, section V concludes, supported by the presented results, and provides perspectives for future works.

II. DEVELOPED APPROACH

A. Framework overview

In order to evaluate perception solutions, it is necessary to have access to reference data. When both reference data and perception data are available for a given scenario, the comparison between data allows us to evaluate the perception solution. The expressed need to be able to evaluate the perception algorithms relies on the availability of reference data.

The proposed framework provides support for offline evaluation of perception algorithms. Thus, perception data are recorded to be replayable and exploitable with our approach. Our aim is to generate reference data without the need of complex classification algorithms or repetitive manual labeling. As it is illustrated in Fig. 1, three data sources compose the proposed methodology : a perception system, a localization system and a 3D map of the environment. They are required for performing the semi-supervised generation of reference data.

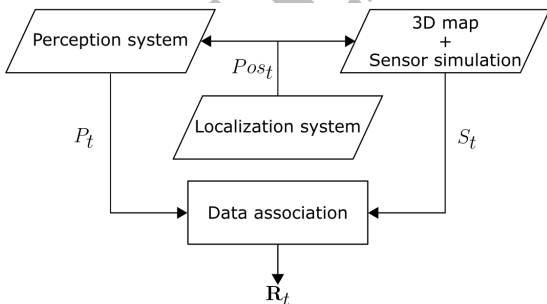


Fig. 1. Pipeline of the reference data generation framework

Perception system: is exploited to record raw perception data P_t acquired under real conditions (e.g. point clouds for LiDAR, images for cameras...), with all the phenomena and external errors. P_t denotes a precise representation of the ego-vehicle surroundings.

Localization system: provides precise localization positions. This information facilitates data association between perception data P_t and model data S_t . To do this, a common reference frame F is exploited for P_t and S_t . Thus, the pose of the perception system recorded at a time t : Pos_t , corresponds to the sensor position and attitude for P_t and is used on the 3D map with the simulator to generate S_t at the same position.

3D map: coupled with a simulation tool allows us to generate model data S_t . The model constituting the georeferenced 3D map was produced using a precise Mobile Mapping System (MMS) Reigl VMX450. Point clouds obtained by MMS were then manually noise-filtered, post-processed, fused and meshed to create the model. Each object label (e.g. road, lane markings, traffic signs...) is associated with a specific id and every part of the mesh constituting the 3D model is labeled in conjunction with the object represented in the scene. This post-processing is performed only once. Subsequently, new datasets recorded in the locations represented by the 3D map will be able to exploit semantic information from this 3D map without requiring new manual labeling.

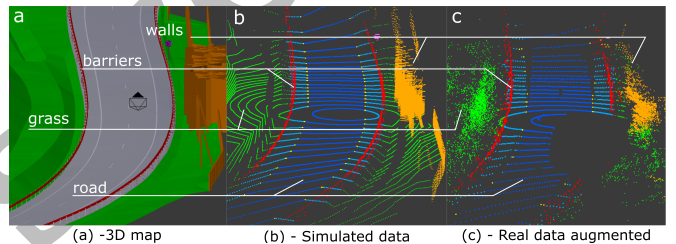


Fig. 2. Comparison between the (a) 3D map, (b) data from the simulated sensor and (c) the real data augmented

3D map is then imported in a settable simulation tool [10], [11], [12]. A perception sensor model with same characteristics of the system used to produce P_t is set and integrated in the simulator. That is, for a given pose Pos_t , data generated by simulation S_t is a model-based perception corresponding to P_t , but contains additional semantic information deduced from the 3D map environment knowledge.

In a last step, a data association method is applied between P_t and S_t in order to transfer semantic environment knowledge contained in S_t on P_t , and outputs reference data R_t . Thus, the combination of these three sources allows us to transfer semantic environment knowledge into the perception data by the way of labeling. Augmented data resulting from this framework represents the reference data and can be exploited in a purpose of evaluation for perception algorithms.

B. System solution and implementation

1) *Perception system:* To record raw perception data P_t , we oriented our system on LiDAR solution. Several reasons motivate this choice. First, it provides range measurements with a centimeter accuracy and each point measured is

independent of other measurements. Second, it is an active sensor, it is not sensitive to light conditions and can be used to record data at any moment of the day without needing modification on the sensor or on its settings. Last, LiDAR data are represented in 3D. Any further treatment on data is necessary to get 3D information as it is the case with point cloud reconstructed from images. Possible errors introduced by the reconstruction step are thus avoided. These advantages grant access to a reliable and detailed representation of the sensed environment in P_t .

A Velodyne VLP-32C is integrated on the top of our vehicle offering the characteristics listed in Table I. P_t produced by this sensor corresponds to 3D point clouds recorded to accurately scan the environment in detail.

TABLE I
PERCEPTION SENSOR CHARACTERISTICS

Velodyne VLP-32C	
Rotation frequency	10 Hz
Points per rotation	20.000
Vertical angular resolution	0.4°
Azimuthal angular resolution	0.1°
Range	200 m
Range measurements accuracy	+/- 3 cm

2) *Sensor simulator*: Bensor [10] was used to generate S_t . This simulation tool is completely settable and provides support to new sensor models. LiDAR sensor model characteristics corresponding to the Velodyne VLP-32C were considered. To generate S_t , 3D map is first imported in the simulator. Once the pose of the sensor is defined, simulated point clouds S_t are obtained by ray tracing the 3D map. Ray-object intersection creates impact points stored in the cloud S_t . Each point is constituted by its 3D coordinates and a label value corresponding to impacted object id.

3) *Localization system*: Finally, the system that plays the link role between P_t and S_t is the localization unit. Positioning accuracy of GNSS solutions in automotive is no better than 2-3 m in good conditions (open sky without outages or multipath effects) [19]. This is not enough to grant access to an accurate data association between S_t and P_t . To perform this task, we oriented our choice on a highly accurate localization solution based on a GNSS-RTK/INS coupling. RTK corrections signals received grant access to centimeter accurate absolute positioning [19], [20], [21]. Coupling with INS (iXblue Atlans-C) gives complementary information of the system attitude. Moreover, high rate of recorded data provided by INS is employed to interpolate the positioning between GNSS-RTK measurements. This configuration outputs an accurate and absolute position of the system.

Positions provided by the localization system are in WGS84 coordinate system. If the 3D map is geo-referenced in a reference frame different from WGS84, transformation is applied on localization data in order to use a common

coordinate system F . In our case, the model is geo-referenced in UTM and we define UTM as our common coordinate system F . By the transformation method described in [22], position coordinates from each pose Pos_t , expressed in WGS84, are projected into UTM coordinate system. Sensor pose Pos_t is used to geo-reference P_t and also to generate S_t . P_t and S_t are geo-referenced with respect to the same pose and overlap each-other allowing the data association.

C. Data augmentation

The last processing step allows us to generate reference data \mathbf{R}_t by a data association between P_t and S_t . The purpose is to augment P_t with semantic environment knowledge included in S_t by transfer of information. In this study, five object classes are taken into account:

$$\mathcal{L} = \{road, signs, borders, lanes, N/A\}$$

To realize the data association, objects represented in both P_t and S_t are matched based on a geometrical criterion. Since we oriented our approach on LiDAR data, P_t and S_t correspond to point clouds. Given the big amount of data that point clouds represent, processing them without optimization can be memory consuming and costly in time. Data association is sped-up using an octree structure [23]. Points are then gathered and stored in voxels. Given a 3D query point q , it is then possible to know in which voxel q is situated, which points compose the same voxel and which ones are located in surroundings voxels. Thanks to this optimization, neighbors research method given a specific a research radius r is much sped-up and is used in Algorithm.1 to outputs \mathbf{R}_t .

Algorithm 1 Data association algorithm

INPUT: P_t - LiDAR point cloud / S_t - Model point cloud
OUTPUT: \mathbf{R}_t - Reference cloud

q - query point
 r - radius of research around q
 n - neighbor found for q
 m - number of neighbors found
 N - set of neighbors found

for all point $q \in P_t$ **do**
 Find the neighbors $n \in S_t$ situated in r
 if $m > 0$, neighbors n are stored in N **then**
 Label values occurrence in N are counted:
 Case 1 - a label value \mathcal{L}_j is more represented than other labels values: \mathcal{L}_j is assigned to q
 Case 2 - several label values have the same number of votes: Euclidian distance d_i is computed between q and each $n_{i \rightarrow m}$, label value of the closest neighbor \mathcal{L}_j is assigned to q
 else
 No neighbor was found in r , q keeps its initial value N/A
 end if
 q is stored in \mathbf{R}_t
end for

According to our approach, S_t is firstly optimized using an octree structure. Each point q from P_t is processed, Algorithm.1 allows us to transfer environment knowledge on P_t and generate reference data \mathbf{R}_t .

Augmented LiDAR data \mathbf{R}_t resulting from Algorithm.1 corresponds to reference data and can then be exploited to evaluate performances of perception solutions.

III. EXPERIMENTAL RESULTS

A. Methodology analysis

1) *Simulation limits:* As stated in Algorithm.1, the data augmentation of P_t depends on the radius of research r . The bigger r , the more neighbors will be found and the more P_t points are labeled. However, the larger the number of found neighbors, the more different label values will be found and important labeling errors can be introduced. Two criteria reflecting these aspects are studied.

The first criterion C_1 reflects the quantity of semantic information introduced by our approach on P_t . To quantify C_1 , we use a sample of 100 reference clouds obtained by our approach. All points contained in P_t are not “labelisable” since objects non-referenced in the 3D map can be captured. This is the case for moving objects (e.g. passage of other vehicles during data recording) or, as we can see Fig.3.a with captured trees and buildings, irrelevant elements distant from the model and non-referenced in the 3D map Fig.3.b.

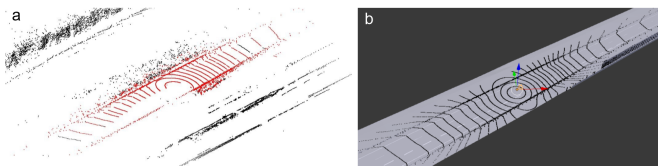


Fig. 3. Comparison between : (a) $-P_t$ with labelisable points (red), non-labelisable points (black), (b) $-S_t$ on the model in Blender

Taking this aspect into account, we compute the number of “labelisable” points by estimating the fitted 3D bounding box of S_t . The number of P_t points contained in this box correspond to the number of “labelisable” points. Finally, C_1 represents the percentage of labeled points compared to the number of “labelisable” points.

The second criterion C_2 reflects errors of labeling caused by the radius of research r . To quantify C_2 , we simulate 100 clouds X_i . X'_i are generated by using the same points stored in X_i and removing semantic labels. Our approach is applied to augment semantic labels on X'_i points. Thereafter, the label value of each point from X'_i is compared to the original value found in X_i . C_2 corresponds to the percentage of wrongly labeled points.

Table.II summarizes the mean values of C_1 and C_2 computed on 100 clouds for five values of r . As reported in Table.II if r is small, risks of wrong labeling is reduced. However, chances to find neighbors in r are also reduced because of geometrical differences between S_t and P_t points position. For further tests, r is set on 0.50m.

TABLE II
COMPUTED CRITERIA C_1 AND C_2 ACCORDING TO r

r (m)	0.10	0.20	0.50	0.80	1.00
C_1	28.80 %	59.73 %	80.73 %	85.27 %	87.38 %
C_2	0.11 %	0.72 %	3.21 %	4.59 %	6.94 %

2) *Role of the localization data:* As explained in II-B3, localization data plays a major role in our approach since the labeling method is based on the geometrical overlap between S_t and P_t . If localization measurements are error impacted or does not reach a sufficient accuracy, offsets between P_t and S_t appear. Therefore, P_t and S_t are not correctly overlapping and labeling errors can appear during data association, affecting the relevance of \mathbf{R}_t . We can notice this aspect in the beginning of one of our datasets, where an irregular offset is visible between S_t and P_t (Fig.4.a), before decreasing until an accurate overlap of the data (Fig.4.b).

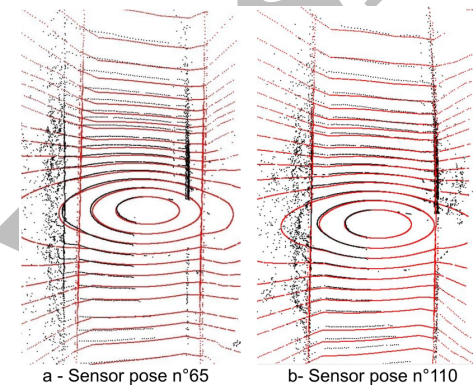


Fig. 4. Offsets between S_t (red) and P_t (black) with : (a) high localization standard deviation, (b) low localization standard deviation.

By studying the standard deviation (σ) of the localization data on position measurements, we can see Fig.5 that the evolution of σ matches perfectly with the noticed offsets. This phenomenon can be explained by an imperfect initialization on the INS causing the high values of σ during the first measurements. Once the INS is fully initialized, the localization measurements become more accurate, σ decreases. The overlapping of S_t and P_t becomes then accurate (Fig.4.b).

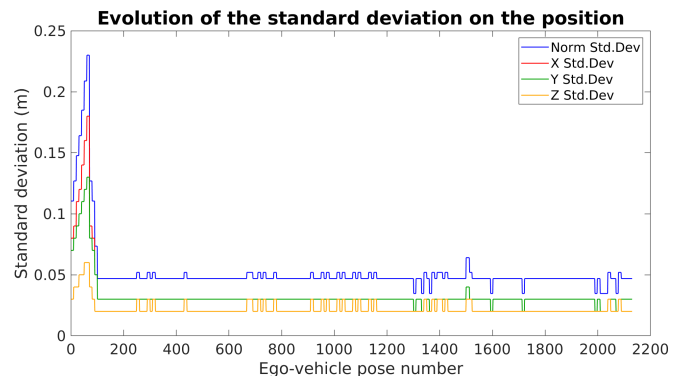


Fig. 5. Evolution of the localization standard deviation on the position

This experiment shows the significance of the localization data in the processing pipeline.

3) *Impact of localization errors*: Localization data accuracy can be affected by measurements errors as it is the case with the initialization of the INS in III-A2. With this experiment, we evaluate impacts of these errors on reference data created. Data used to lead the test are generated by simulation tool [10].

- First, we use a sequence of 100 sensor poses Pos_i and we simulate the 100 corresponding clouds S_i .
- Second, we introduce centered -additive white Gaussian noise on localization data. Noised poses are named Pos'_i . 100 clouds X_i are simulated with noisy poses Pos'_i but geo-referenced with Pos_i . This way, errors on localization data can be simulated and controlled according to the noise introduced. X_i generated by the simulator are correctly labeled.
- A copy X'_i is made and label value of each point is re-initialized with “N/A”. X'_i is then labeled by our approach with S_i .
- Finally, we compare each point between X_i and X'_i and evaluate the percentages of points badly labeled.

Fig.6 represents effects of localization data noise on clouds labeled by our approach. The experiment is leaded several levels of noise introduced in Pos'_i from 0.10m to 5m.

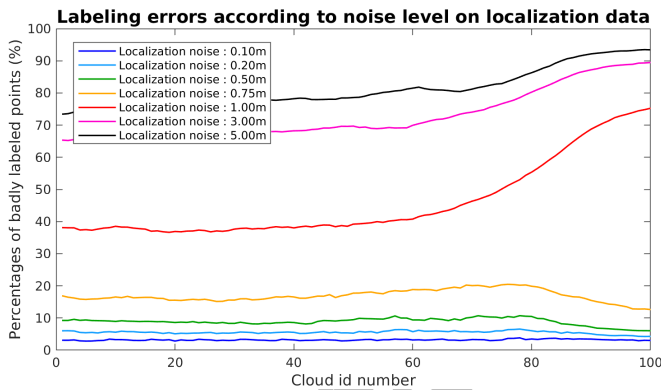


Fig. 6. Labeling errors according to noise level on localization data

The results shown in Fig.6 demonstrates that the accuracy of localization data has to be carefully taken into account in the process. The impact of localization noise is accentuated in turns as we see with increasing curves. Thus, if localization accuracy indicators show a low confidence level, data resulting from our approach is not exploitable as reference data.

B. Application of the approach

In this last experiment, performances of a 3D road detection algorithm [24] are evaluated thanks to reference data generated by our framework. It is worth noting that such an evaluation is suitable for algorithms and sensors performing in 3D and 2D. To operate this test, a sequence of 200 clouds is exploited. Fig.7 illustrates the processing pipeline including the road detection.

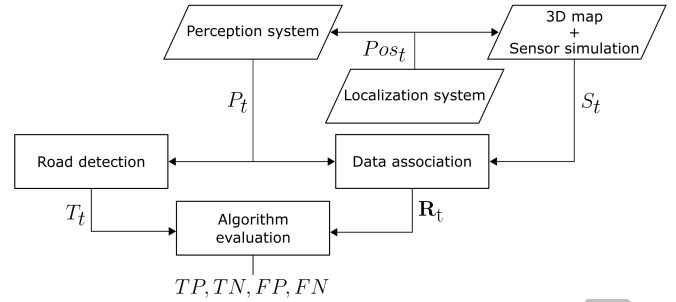


Fig. 7. Steps of the road detection algorithm evaluation

Road Detection : Road detection algorithm performs LiDAR point cloud P_t . All the points that are detected as road receive the label value “road”. Remaining points keep their initial label value “N/A”. The outputs labeled cloud tested is named T_t .

Algorithm Evaluation : The last step of our approach receives both T_t and R_t to evaluate the processing applied, in our case, the road detection algorithm. Both clouds are composed of the same points, only the label value can be different between peer points. Evaluation metrics : true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are computed by comparing points label values \mathcal{L} as detailed in Algorithm.2.

Algorithm 2 Evaluation metrics computing

INPUT: R_t - Reference cloud / T_t - Tested cloud

OUTPUT: TP, TN, FP, FN - Evaluation criteria

for all point $p_i \in T_t$ **do**

Find the peer point $r_i \in R_t$

if \mathcal{L}_{p_i} and $\mathcal{L}_{r_i} = \text{road}$ **then**

TP = + 1

else if \mathcal{L}_{p_i} and $\mathcal{L}_{r_i} \neq \text{road}$ **then**

TN = + 1

else if $\mathcal{L}_{p_i} = \text{road}$ and $\mathcal{L}_{r_i} \neq \text{road}$ **then**

FP = + 1

else if $\mathcal{L}_{p_i} \neq \text{road}$ and $\mathcal{L}_{r_i} = \text{road}$ **then**

FN = + 1

end if

end for

Thanks to TP, FP and FN criteria, we can deduce standard metrics recall and precision for each cloud of the sequence. On Fig.8, curves corresponding to standard metrics for each cloud are represented. By analyzing curves, we are able to get a detection performances overview of the tested algorithm and its behavior overall the sequence. Peaks allows us to quickly identify parts of sequence where the performances decrease.

Investigations are then facilitated (Fig.9) to identify why the efficiency of the tested algorithm varies, what are the causes responsible of performance decreases and how the algorithm can be improved to overcome these causes.

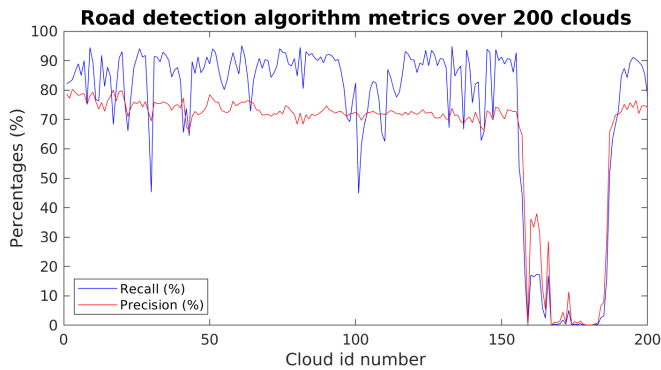


Fig. 8. Evaluation metrics curves

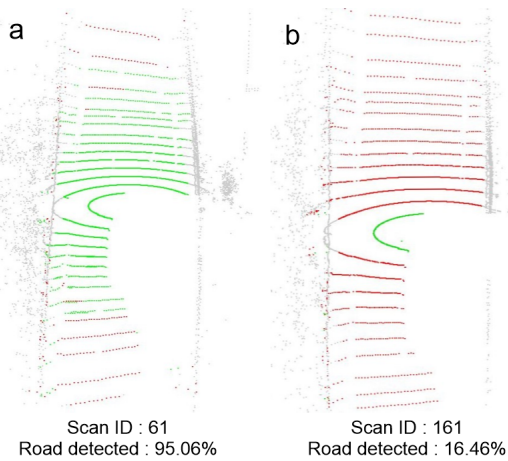


Fig. 9. Comparison of two cases among the algorithm results : (a) Accurate road detection rate. (b) Inaccurate road detection rate. (Green points = points well classified after comparison to our referencing approach, Red points = points missed after comparison to our referencing approach)

IV. CONCLUSIONS

In this paper, a semi-supervised reference data generation method is presented. The joint use of LiDAR data with synthetic data generated from a detailed 3D map, all linked by localization data is exploited to augment semantic information in LiDAR clouds and generate reference data. Transfer of environment knowledge stored in synthetic data on LiDAR data allows us to avoid the need of complex classification algorithms. Thus, labeling errors are greatly reduced. However, sensibilities on the localization accuracy and the neighbors search parameters must be carefully taken into account.

As a perspective, improving the simulated data to make them closer to LiDAR data will allow us to reduce radius of research. Risks of labeling errors will be reduced while keeping a high percentage of transferred semantic information. To deal with localization accuracy variations, clouds alignment methods could also be introduced to support low confident localization measurements. Efforts will be done to evaluate other sensor modalities (e.g. vision systems) and to efficiently manage dynamic objects on the evaluated scenarios.

REFERENCES

- [1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [2] Y. Xu, Z. Sun, R. Boerner, T. Koch, L. Hoegner, and U. Stilla, "Generation of ground truth datasets for the analysis of 3d point clouds in urban scenes acquired via different sensors," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018.
- [3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [4] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [5] T. Tatschke, F. Farber, E. Fuchs, L. Walchshausl, and R. Lindl, "Semi-autonomous reference data generation for perception performance evaluation," in *International Conference on Information Fusion*, 2007.
- [6] A. Borkar, M. Hayes, and M. T. Smith, "A novel lane detection system with efficient ground truth generation," *IEEE Transactions on Intelligent Transportation Systems*, 2012.
- [7] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [8] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Neural Information Processing Systems*, 2015.
- [9] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European Conference on Computer Vision*, 2008.
- [10] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender sensor simulation toolbox," in *Advances in Visual Computing*, 2011.
- [11] S. Bechtold and B. Hofle, "Helios : a multi-purpose lidar simulation framework for research, planning and training of laser scanning operations with airborne, ground-based mobile and stationary platforms," *Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016.
- [12] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning*, 2017.
- [13] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [14] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," *Computing Research Repository*, 2017.
- [15] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" *Computing Research Repository*, 2016.
- [16] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kotschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *International Conference on Computer Vision*, 2017.
- [17] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *International Journal of Computer Vision*, 2018.
- [18] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A lidar point cloud generator: from a virtual world to autonomous driving," *International Conference on Multimedia Retrieval*, 2018.
- [19] J.-M. Zogg, *GPS Essentials of Satellite Navigation Compendium*, 2009.
- [20] Y. Feng and J. Wang, "GPS RTK performance characteristics and analysis," *Journal of Global Positioning Systems*, 2008.
- [21] B. Scherzinger, "Precise robust positioning with inertial/gps rtk," *ION-GPS*, 2000.
- [22] J. P. Snyder, *Map projections—A working manual*, 1987.
- [23] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the Association for Computing Machinery*, 1975.
- [24] S. Rodriguez, V. Fremont, P. Bonnifait, and V. Cherfaoui, "An embedded multi-modal system for object localization and tracking," in *IEEE Intelligent Vehicles Symposium*, 2010.