2018 42nd IEEE International Conference on Computer Software & Applications

Machine Learning Assisted High-Definition Map Creation

Jialin Jiao
Uber Technologies, Inc.
San Francisco, U.S.A.
jiaojialin@gmail.com

***Abstract*— In recent years, autonomous driving technologies have attracted broad and enormous interests from both academia and industry and are under rapid development. High-Definition (HD) Maps are widely used as an indispensable component of an autonomous vehicle system by researchers and practitioners. HD Maps are digital maps that contain highly precise, fresh and comprehensive geometric information as well as semantics of the road network and surrounding environment. They provide critical inputs to almost all other components of autonomous vehicle systems, including localization, perception, prediction, motion planning, vehicle control etc. Traditionally, it is very laborious and costly to build HD Maps, requiring a significant amount of manual annotation work. In this paper, we first introduce the characteristics and layers of HD Maps; then we provide a formal summary of the workflow of HD Map creation; and most importantly, we present the machine learning techniques being used by the industry to minimize the amount of manual work in the process of HD Map creation.**

Keywords-High-Definition Map, HD Map Creation, Autonomous Vehicle System

I. INTRODUCTION

Fully autonomous vehicle systems had never been closer to reality as they are today: thousands of them are being tested daily on the roads around the world [53,54]; experts have predicted that between 2020 to 2040, autonomous vehicles will become very normal in our life and we will be seeing more and more autonomous vehicles running within the current traffic systems, self-driving alongside with human-driving cars, cyclists and pedestrians [1].

High-Definition (HD) Maps for autonomous vehicles are pre-built digital model of the driving environment with highly precise, fresh and comprehensive geometric information and semantics. As early as in DARPA challenges in the 2000s, HD Maps have already been used for precise localization of the autonomous vehicles [2,3,6,60]. However, the usefulness of HD Maps is more than just for precise localization. To build a fully autonomous vehicle system operating in real-world environments (most challengingly, in urban environments [3,6,59]), many components need to work closely together and be optimized holistically. This task is so complicated with enormous inputs, parameters and uncertainty that it is very challenging to do all the computation in real time while still meeting the performance and safety requirement; also, some static elements and properties of the driving environment might be difficult to detect by sensors reliably and efficiently at runtime. To assist with those, HD Maps capture other useful prior information besides what is needed for localization and store the result of pre-computation for many other problems autonomous vehicles needs to solve, including perception, prediction, motion planning, vehicle control etc. [51,57,59]. One example of such pre-computation is the mapping of the 3D locations of the traffic lights, which allows autonomous vehicles to only examine a small region instead of the whole field of view to efficiently detect the state of a traffic light [13]. While there are debates about the possibility of building a fully autonomous vehicle system without using pre-built HD Maps, no existing highly automated driving (HAD) systems we know of are running in urban environments without using some kind of HD Map.

Historically, it is a complex and mostly-manual or semi-automated process to build HD Maps, requiring quite a wide range and significant amount of software and manual effort [22,51,57,64]. It is especially laborious and costly to extract semantics from data. Automation of such manual work is critical to improve the efficiency of the process and the quality of the HD Maps. Heuristics based (e.g. [25,27-30]) and machine learning based approaches are both used for such automation. This paper will be focused on examining the use of machine learning techniques to assist with the creation of HD Maps. Also, while recently some companies are trying to create HD Maps from camera images only [48,57,58], the HD Map creation process discussed here uses both LiDAR point clouds and camera images. It is not intended to be an exhaustive survey in every aspects, but we are hoping it could help to attract more and more interests in applying machine learning techniques in this area.

The paper is organized as follows: section II briefly introduces the basic concepts of HD Maps, section III talks about the general workflow of HD Map creation, section IV and V discuss the reasons of applying machine learning techniques in HD Map creation and an aspect of this kind of machine learning, followed by a review of examples of machine learning applications in different steps in section VI; lastly, section VII reviews some recent advances on end-to-end deep learning directly on 3D point clouds.

II. BASICS OF HIGH-DEFINITION MAP

HD Maps for autonomous vehicles are different from regular digital maps (web based or mobile based) used by us human and have the following special characteristics.

A. Characteristics of High-Definition Maps

HD Maps for autonomous vehicle systems should have:

0730-3157/18/$31.00 ©2018 IEEE
DOI 10.1109/COMPSAC.2018.00058

1) High Precision

As the name suggests, HD Maps for autonomous driving systems need to have a high precision, usually at centimeter-level. While there is no standard about what exactly the precision should be, it is common in the community to see HD Maps with precision between 5 ~ 20 cm [6,13,57,59,60], the implication of which is that an autonomous vehicle using such a HD Maps for localization could potentially localize itself within 5 ~ 20 cm margin of error.

2) Rich Geometric Information and Semantics

HD Maps must contain rich geometric and semantic information of the road network and surrounding environment for use by localization, perception, prediction, motion planning and vehicle control etc. The most common content includes lane/road model, 3D locations of traffic control devices (mainly traffic lights and traffic signs), and geometry and semantics of other static road elements such as curbs, crosswalk, railway tracks, guardrails, poles, bus stops, speed bumps, potholes, overpass etc.

3) Fresh Data

HD Maps need to be updated with changes timely. TomTom estimates about 15% of US roads change every year in some ways [4]; although not all of those changes are of concern for autonomous vehicles, we could infer the order of magnitude of relevant changes that need to be updated to ensure the safety of all parties on the roads where autonomous vehicles are operating.

B. Layers of High-Definition Maps

HD Maps usually have multiple layers and together they provide a full stack of information for autonomous vehicles. Because of the size of all the layers, they are usually being served to autonomous vehicles from the cloud [5,41], and only a few nearby small areas of the HD Map (called submaps) are downloaded to the vehicle when needed.

Layers of HD Maps are quite different from each other and have different representations, data structures and purposes. Although HD Maps builders do not necessarily follow the same practice, HD Maps usually contain the following 4 layers [55,56]:

1) 2D Orthographic Reflectivity Map

Orthographic reflectivity map leverages the fact that different materials (e.g. different types of road pavement, road marking paints etc.) on the road surface have different infrared reflective intensity from laser. This layer is a 2D planar view (bird's-eye view) of the road surface extracted from the LiDAR 3D point clouds. The reflectivity map actually may look photorealistic after combining multiple scans of the same area and texturing the intensity value on to the points. Reflectivity maps are mostly useful for localization [6,7,57,59].

2) Digital Elevation Model (DEM)

DEM is a 3D model and models the 3D shape of the surface of the driving environment, such as the height of the road curbs, the grade/steepness of a ramp or hilly road etc. It is useful for localization, perception, motion planning and vehicle control [55,57]. Examples of DEM visualization could be found at [8,55,56].

3) Lane/Road Model

Lane/road model is a very important vectorized layer that contains the semantics of lane segments and road segments. Road model includes the parts of road that aren't part of the lanes, such as the edge of the road. However, since we will always try to center the autonomous vehicles in the lane most of the time, in reality autonomous vehicles mostly only need to deal with the lane model unless in rare occasion they need to travel outside of the lane boundary. Lane model should contain information of lane geometrics (boundaries, width, curvature etc.), lane type (car lane, bike lane, bus-only lane etc.), lane directionality, lane marking/divider types (solid vs dashed, single vs double etc.), restrictions (e.g. left/right turn only), speed limits, connectivity between lanes, etc. [57] Lane/road model is critical for localization [59], motion planning, vehicle control etc.

4) Stationary Map

This is usually a versatile layer that stores the semantics of those static elements in the driving environment and pre-computations that are not captured in other layers (e.g. traffic lights and their association with lanes [13], road obstacles, optimal path when traffic permits [51] etc.)

III. HIGH-DEFINITION MAP CREATION PROCESS

The HD Map creation process could be broken down into 4 stages: data collection, HD Map generation, quality control and validation, update and maintenance (Fig. 1).

A. Data Collection

Mobile Mapping Systems (MMS) equipped with LiDAR scanner(s), cameras, GPS, IMU (Inertial Measurement Unit), Wheel Odometer etc. go on field trips to collect data and log them into hard drives (or send the data to data center or cloud storage via cellar network after some kind of processing, filtering and compression [52]). The data collection is usually carried out by zones of a city and involves carefully routes planning, standardized operation procedural, optimized data storage and transmission etc. Cost of equipment, human labor, data storage and transmission are the major concern for data collection; and reducing the times of driving on same road segment is of great interest to practitioners.

The data collected for HD Map creation belong to two categories: (1) HD Map data: the LiDAR point clouds and camera images contain the geometry and semantic information that will become content of the HD Maps; a comparison of point clouds and images could be found in Table 1; Note that not all HD Maps makers use LiDAR point clouds; some only use camera images and do 3D reconstruction from them [48,57,58], which is out of the scope of this paper. (2) Auxiliary data: these include the log of GPS/IMU/Wheel Odometer that are useful for generating the HD Maps, but do not transform to HD Maps content. The use of the auxiliary data is mainly for precisely estimating the pose of the data collecting vehicles which we will discuss shortly.

TABLE I. COMPARISON OF POINT CLOUDS AND IMAGES

Data	Characteristics	Uses in HD Map Creation
Point Clouds	3D, precise, independent of illumination, could be noisy, sparse, lacking texture and color	3D location detection directly, geometry extraction, some semantics/attributes extraction
Images	2D, high resolution, quality affected by lighting conditions	3D location detection through triangulation, 3D reconstruction, Semantics/attributes extraction

B. HD Map Generation

This is the back-office work that processes the collected data and generates the HD Maps. Roughly it could be further broken down into 4 steps (Fig. 1):

1) Sensor Fusion and Pose Estimation

Knowing the accurate pose (location and orientation) of a data collection vehicle is key to generating HD Maps. If the poses of the vehicles are inaccurate, it is impossible to produce precise maps. Once we have the accurate poses of the vehicle, and given we know where the sensors are mounted and their relative angles to the vehicle frame, we could infer the accurate pose of the point cloud and image frames easily.

Although accurate pose could not be acquired directly at runtime due to the limitation of GPS, IMU and wheel odometry etc. [59] (unless performing online localization against a pre-built HD Maps, but we do not have HD Maps yet), accurate pose could be estimated by offline optimization by fusing log of different sensors using graph-based SLAM (Simultaneous Localization and Mapping) [6,9].

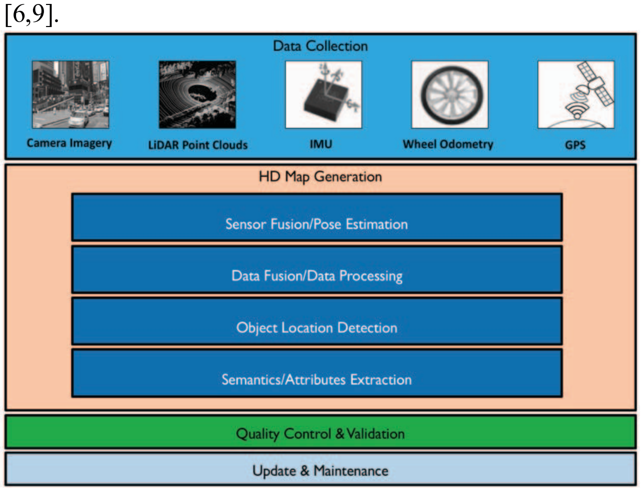

Figure 1. Workflow of HD Map Creation

2) Map Data Fusion and Data Processing

Once we have accurate poses, we then could do map data (LiDAR point clouds and camera images) fusion [50]. Note that for HD Mapping, the resolution and quality of videos are usually not satisfactory, and we don't need such a high frame rate as from videos, so higher resolution still images taken a few frames per second (usually below 10 frame/sec) are commonly used [13]. During the data fusion, multiple scans of point clouds are aligned and calibrated to get denser point clouds; and point clouds and camera images are registered to each other so that we could use the point cloud to get the 3D location of objects directly and the registered images to recognize the semantics because point clouds provide 3D position but usually too sparse for sign content while images will do a great job on those but do not provide 3D information.

Other data processing work are also carried out including road plane generation, removal of irrelevant objects (e.g. dynamic objects and objects too far away from the road), and texturing to generate photorealistic orthographic images etc.

3) 3D Object Location Detection

For road elements whose precise geometry and location are important (e.g. lane boundaries, stop lines, curbs, traffic lights, overpasses, railway tracks, guardrails, light poles, speed bumps, even potholes etc.), we need to map their precise 3D locations. LiDAR point clouds contain 3D location information and 3D object detection on point clouds are performed either using geometry-based method [27-30] or deep learning on 3D point clouds [10-12]. We could also detect 3D object locations by triangulation from images, one such example could be found in [13].

4) Semantics/Attributes Extraction

The last and also the step with most work is to extract semantics and attributes from data. The work usually includes: lane/road model construction, traffic signs recognition and association with lanes, association of traffic lights with lanes, road marking semantics extraction, various road elements (e.g. light poles) detection etc.

There are actually other works need to be done before a large-scale HD Map could be generated, but the aforementioned steps are the major ones.

C. Quality Control and Validation

Once HD Maps are generated, pre-defined quality metrics must be met, and HD Maps could be validated by different means including testing on road and verification by using other survey methods.

D. Update and Maintenance

This stage is the continuous work to keep the HD Maps updated timely and fix issues discovered during the use of them.

IV. MACHINE LEARNING ASSISTED HD MAP CREATION

The following aspects of HD Map creation process have made machine learning techniques a natural choice to improve the efficiency of HD Map creation and quality of the HD Maps:

A. Significant human labor required

Months and years of hundreds or even thousands of people are needed to build large-scale HD Maps manually or semi-automatically, and much of the manual work are repetitive, tedious; not just they are time consuming and costly, but also prone to error because the tasks require great deal of attention and focus. Because of this aspect, there is no better choice than implementing automation by software, and machine learning has proven to be able to do a great job in many of such manual tasks.

B. Massive data with high-dimension

Whenever there is a lot of data and data has high dimensions, the tool we usually seek for help is machine learning techniques.

C. Shared problems with other autonomous driving tasks

Many of the problems we need to solve for building HD Maps overlap with the problems from perception, localization, prediction etc. Since there are already a lot of machine learning work done for those tasks, it is wise to borrow some of those tools and algorithms to use for HD Map creation.

There are actually other reasons to apply machine learning in HD Map creation, including the argument that machine learned model are easier to generalize and be transferred to solving similar problems in other cities or regions.

V. "Human-in-the-loop" Machine Learning

The way machine learning techniques are used in HD Map creation has a very evident characteristics that it is commonly understood as a "human-in-the-loop" machine learning [42-44,46]. "Human-in-the-loop" machine learning in HD Map creation is the process of iteratively improving the machine learning models during the process of creating the HD Maps by acting on feedback to the results of the models with the involvement of human (including operators for annotation and researchers/engineers for training and improving the machine learning models). "Human-in-the-loop" machine learning is especially useful when the tasks require very high accuracy, but the performance of the machine learning model isn't there yet. As shown in Fig.2, in the very beginning, there is no machine learned models, all we have is just unlabeled data collected by the MMS vehicles; trained operators will have to manually label all the data and generates HD Maps; now that we have labelled data, researchers/engineers could train supervised machine learning models which are used to classify remaining or newly collected unlabeled data. There is a critical ingredient that we must have to make the "human-in-the-loop" machine learning work: confidence estimation of the machine learning model predictions (while the topic of estimating the confidence of the output of different types of machine

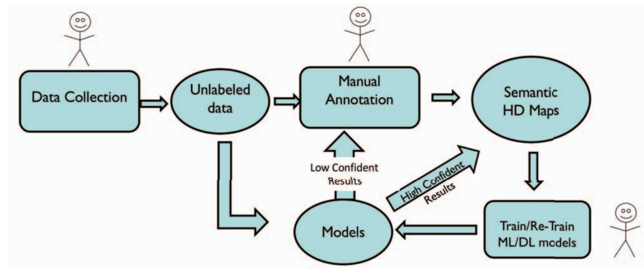

Figure 2. "Human-in-the-loop" Machine Learning in HD Map Generation

learning models is out of the scope of this paper, a few methods for estimating confidence scores of neural network predictions are worth mentioning, including Bayesian approach using Monte Carlo dropout [14], entropy based confidence score with Adversarial Training [15], and distance-based confidence score [16]). Once we have the confidence scores of the machine learned model output, we could save the high-confident output into the HD Maps directly and send the low-confident output to human operators to judge, and the manual labels for those low-confident results then go into the HD Maps and are fed back to re-train the machine learning models in next iteration. In some sense, "human-in-the-loop" machine learning is a type of active learning [17].

VI. Examples Of Machine Learning Uses in HD Map Creation

In this section we are going to examine the applications of machine learning techniques in some of the HD Map creation steps. Note that the effectiveness of machine learning in assisting HD Map creation mainly lies in semantics/attributes extraction.

A. Pose Estimation

Traditionally, pose estimation could be done with techniques such as filter-based [38,39] or graph-based SLAM[37] [37] or visual SLAM [40]. Recently there are deep learning based approaches to tackle the pose estimation problem using images [18,19]; one example is the PoseNet [18], which is a modified GoogLeNet for estimating 6-DOF (Degree of Freedom) pose. It replaces the softmax classifiers with affine regressors and outputs a pose vector at the final fully connected layer. Although the performance of the PoseNet is not satisfactory for use for making HD Maps, it is after all a first step in the direction toward applying deep learning in solving the pose estimation problem.

B. Lane/Road Marking Extraction

Lane/Road marking extraction is the extraction of semantics from the markings on the road surface. Lane/road markings are markings painted on the road surface, including lane boundaries, lane divider lines, arrows, crosswalk (zebra crossing) markings, speed limit texts, lane type markings etc. Quite some work has been done in extraction of lane/road markings from LiDAR point cloud: a lot of them use heuristics and rule-based approaches, and we will only discuss those using machine learning in some way. [20] uses Deep Boltzmann Machine to classify small-size road markings (arrows and rectangles) after extracting them from road surface points using reflectivity thresholds, and then uses PCA to further differentiate crosswalk (rectangles stacked vertically) from dashed lane line (rectangles lined up horizontally); the authors reported a completeness (i.e. recall) of 93%, and correctness (i.e. precision) of 92%. Baidu uses a CNN to extract from reflectivity imaging generated from point clouds, and by using multiple deconvolution layers with un-pooling, the resulting HD Maps could reach impressive resolution (up to 1cm x 1cm precision) and the pixel level recall and precision of their method are 93.80% and 95.49% [21].

C. Traffic Light Mapping

Traffic Light mapping is to map the location of the traffic lights and associate them with corresponding lanes so that it could speed up the traffic light state detection for autonomous vehicles in run-time. Google maps traffic lights by using an image-based method. They first get accurate poses either by offline optimization with SLAM or online localization with a pre-built HD Maps, then they filter out most images that won't contain traffic lights according to their proximity to intersections, then a machine learned classifier is used to detect the traffic lights in the images and an iterative process is used to triangulate the 3D location from multiple images of the same traffic lights and associate images groups by same 3D locations until converge. Results show their method could map 95% ~ 99% of the traffic lights with location error less than 15 cm [13].

D. Traffic Sign Mapping

Traffic sign mapping are usually done in two ways:

1) Image-based method similar to the way of traffic light mapping aforementioned;

2) Method based on fused images-point clouds data: Basically, the idea is to first detect the location of the traffic signs using 3D point clouds and recognize the semantics of the signs using registered images (this could be done easily by CNN with great performance nowadays).

[22] uses SVM to detect all types of traffic signs from 3D point clouds directly (without reading their content) with a precision of 89%, but their work does not recognize the content of the signs (except they detect stop sign by their unique shape).

E. Road Edge/Curb Extraction

Road edge/curb extraction is a task for building vectorized lane/road model. Traditionally, this task has been done by operators manually outlining the edge/curb with visualization tools. There are also automations mainly based on heuristics of elevation difference of points of the curbs and the road surface [23,24], some also use heuristics based on density and slope changes [23] or other heuristics [25]. Some works using deep learning has been published but the performance is not there yet [26].

F. Light Poles Extraction

Light poles are the kind of road elements that could help with localization, especially in time when there is no much features from road surface. Historically a lot of work in detecting light poles are done with geometry-based heuristics or using energy function, e.g. [27]. Works using machine learning also show good performance; usually machine learning models are used in classification of the pole-like objects after candidates are being identified by geometry-based methods. For example, [28] uses a Gaussian Mixture Model to recognize the lighting pole from candidates, the overall performance reaches a true positive rate of 90%; [29] use linear discriminant analysis and support vector machines to classify different types of pole-like objects including lighting pole, reaching accuracy over 90%. [30] uses random forest to classify pole-like objects and the precision and recall for light pole reach 94.8% and 97.5%.

G. HD Map Refresh/Update

In order to refresh HD Map timely, at least two things need to happen: collecting fresh HD data timely and detecting changes from the fresh data. To collect fresh data, it is usually not cost-efficient to send the mapping vehicles out everywhere to re-collect data constantly [52]. While autonomous driving companies could rely on their autonomous vehicles being tested on road to collect fresh data with high quality, the coverage is still in question; Some HD Maps suppliers work with automakers to get fresh map data from collected intelligent vehicles equipped with various sensors [63]; currently many HD Maps builders are also taking the path of crowdsourcing [49] where the challenges lie in ensuring the quality of the data meet the need of HD Maps. Once fresh data is in, road changes detection and road events (e.g. road closure) detection must be done. The problem of training a machine learning model to detect road events from images is that training data for road events might be comparatively rare (data imbalance between positive instances and negative instances). To solve this, usually transfer learning [31] is used to leverage pre-trained models. One might also try to use aggregated GPS traces from connected vehicles or mobile devices to detect changes on traffic pattern to infer the possibility of road changes. [45,47,52,61,62].

VII. Deep Learning on 3D Point Cloud

Although LiDAR scanners are quite costly today, they are still the primary sensor for HD Mapping mainly because many benefit of LiDAR point clouds (see Table 1). As deep learning has become dominant in computer vision from images, researchers and practitioners had begun to explore applying deep learning to 3D point clouds directly.

There are many challenges to learn from point clouds directly due to the characteristics of point clouds:

(1) Different from pixels in 2D images, points in point clouds are unordered and unstructured;

(2) Points are sparse;

(3) Highly variable point density;

(4) Point cloud data is noisy: missing data is common;

(5) Lacking color and texture;

(6) Misalignments due to vehicle motion etc.

Historically, when doing machine learning on point clouds data, point clouds are first converted to other representation (e.g. voxelization [32-34] or projection into a perspective view [32,35]) before handcrafted feature engineering for specific tasks. There is hardly any work that could train a more generic machine learning model end-to-end directly from point clouds data until the proposals of PointNet, PointNet++, VoxelNet [10-12] etc.

PointNet [10] is a deep network architecture that could be trained end-to-end from point clouds directly. Point cloud is input to the network as an N (number of points) by D (dimension) 2D matrix. PointNet is proved to be pretty robust when there is data corruption. The weakness of PointNet is that it can't learn local structure thus hard to be generalized to large scale scene. It has a classification network and a segmentation network whose performance is on par or slightly better than previous deep networks learned from point clouds after they are converted to other forms of representation.

PointNet++ [11] improves on PointNet. It is a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. The hierarchical point set feature learning is analogous to the multiple layers of convolution operations of a ConvNet: PointNet++ extracts local features capturing fine geometric structures from small neighborhoods; such local features are further grouped into larger units and processed to produce higher level features. The 3D shape classification accuracy of PointNet++ reach 91.9% when tested on ModelNet40 data set.

VoxelNet [12] is introduced by Apple and is another end-to-end trainable deep neural network architecture on 3D point clouds directly. It proposes a novel voxel feature encoder to transform point clouds into descriptive volumetric representation before feeding into an RPN (region proposal network) to generate detection. VoxelNet was once the leading classifier in KITTI car detection benchmark [36].

VIII. Conclusion

The contribution of this paper is two-fold: first, it provides a first of a kind (to the best of my knowledge) of formal summary of the complete workflow of HD Map creation process used in the industry; second, it presents a detailed review of machine learning techniques used in assisting the creation of HD Maps by industry practitioners as well as academic researchers. Hopefully this paper will be informative and interesting to audience who work in HD Map creation and autonomous vehicles systems and to those who care about improving the efficiency of HD Map creation and the quality of the HD Maps.

References

[1] Liu, S., Peng, J. and Gaudiot, J.L., 2017. Computer, Drive My Car! Computer, 50(1), pp.8-8.

[2] Buehler, M., Iagnemma, K. and Singh, S. eds., 2007. The 2005 DARPA grand challenge: the great robot race (Vol. 36). Springer Science & Business Media.

[3] Buehler, M., Iagnemma, K. and Singh, S. eds., 2009. The DARPA urban challenge: autonomous vehicles in city traffic (Vol. 56). springer.

[4] https://automotive.tomtom.com/wordpress/wp-content/uploads/2017/01/Brochure-TomTom-Automotive-1.pdf

[5] Liu, S., Li, L., Tang, J., Wu, S. and Gaudiot, J.L., 2017. Creating Autonomous Vehicle Systems. Synthesis Lectures on Computer Science, 6(1), pp. i-186.

[6] Levinson, J., Montemerlo, M. and Thrun, S., 2007, June. Map-Based Precision Vehicle Localization in Urban Environments. In Robotics: Science and Systems (Vol. 4, p. 1).

[7] Automatic laser calibration, mapping, and localization for autonomous vehicles, Jesse Levinson. Thesis (Ph.D.), Stanford University, 2011

[8] https://medium.com/waymo/building-maps-for-a-self-driving-car-723b4d9cd3f4

[9] Thrun, S. and Montemerlo, M., 2006. The graph SLAM algorithm with applications to large-scale mapping of urban structures. The International Journal of Robotics Research, 25(5-6), pp.403-429.

[10] Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 1(2), p.4.

[11] Qi, C.R., Yi, L., Su, H. and Guibas, L.J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems (pp. 5105-5114).

[12] Zhou, Y. and Tuzel, O., 2017. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. arXiv preprint arXiv:1711.06396.

[13] Fairfield, N. and Urmson, C., 2011, May. Traffic light mapping and detection. In Robotics and Automation (ICRA), 2011 IEEE International Conference on (pp. 5421-5426). IEEE.

[14] Gal, Y. and Ghahramani, Z., 2016, June. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning (pp. 1050-1059).

[15] Lakshminarayanan, B., Pritzel, A. and Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems (pp. 6405-6416).

[16] Mandelbaum, A. and Weinshall, D., 2017. Distance-based Confidence Score for Neural Network Classifiers. arXiv preprint arXiv:1709.09844.

[17] Krishnakumar, A., 2007. Active learning literature survey. Technical Report, University of California, Santa Cruz.

[18] Kendall, A., Grimes, M. and Cipolla, R., 2015, December. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Computer Vision (ICCV), 2015 IEEE International Conference on (pp. 2938-2946). IEEE.

[19] Walch, F., Hazirbas, C., Leal-Taixé, L., Sattler, T., Hilsenbeck, S. and Cremers, D., 2016. Image-based localization using LSTMs for structured feature correlation. arXiv preprint arXiv:1611.07890.

[20] Yu, Y., Li, J., Guan, H., Jia, F. and Wang, C., 2015. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(2), pp.709-726.

[21] He, B., Ai, R., Yan, Y. and Lang, X., 2016, November. Lane marking detection based on Convolution Neural Network from point clouds. In Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on (pp. 2475-2480). IEEE.

[22] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J.Z., Langer, D., Pink, O., Pratt, V. and Sokolsky, M., 2011, June. Towards fully autonomous driving: Systems and algorithms. In Intelligent Vehicles Symposium (IV), 2011 IEEE (pp. 163-168). IEEE.

[23] Zhou, L. and Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. International Journal of Applied Earth Observation and Geoinformation, 18, pp.293-304.

[24] Zhang, W., 2010, June. Lidar-based road and road-edge detection. In Intelligent Vehicles Symposium (IV), 2010 IEEE (pp. 845-848). IEEE.

[25] Yang, B., Fang, L. and Li, J., 2013. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 79, pp.80-93.

[26] Rachmadi, R.F., Uchimura, K., Koutaki, G. and Ogata, K., 2017, September. Road edge detection on 3D point cloud data using Encoder-Decoder Convolutional Network. In Knowledge Creation and Intelligent Computing (IES-KCIC), 2017 International Electronics Symposium on (pp. 95-100). IEEE.

[27] Yu, Y., Li, J., Guan, H., Wang, C. and Yu, J., 2015. Semiautomated extraction of street light poles from mobile LiDAR point-clouds. IEEE Transactions on Geoscience and Remote Sensing, 53(3), pp.1374-1386.

[28] Zheng, H., Wang, R. and Xu, S., 2017. Recognizing Street Lighting Poles from Mobile LiDAR Data. IEEE Transactions on Geoscience and Remote Sensing, 55(1), pp.407-420.

[29] Ordóñez, C., Cabo, C. and Sanz-Ablanedo, E., 2017. Automatic Detection and Classification of Pole-Like Objects for Urban Cartography Using Mobile Laser Scanning Data. Sensors, 17(7), p.1465.

[30] Fukano, K. and Masuda, H., 2015. Detection and Classification of Pole-like Objects from Mobile Mapping Data. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 2.

[31] Pan, S.J. and Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), pp.1345-1359.

[32] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2016.

[33] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In IEEE/RSJ International Conference on Intelligent Robots and Systems, September 2015.

[34] D. Z. Wang and I. Posner. Voting for voting in online point cloud object detection. In Proceedings of Robotics: Science and Systems, Rome, Italy, July 2015.

[35] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. In Robotics: Science and Systems, 2016.

[36] http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d

[37] Grisetti, G., Kummerle, R., Stachniss, C. and Burgard, W., 2010. A tutorial on graph-based SLAM. IEEE Intelligent Transportation Systems Magazine, 2(4), pp.31-43.

[38] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. and Leonard, J.J., 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics, 32(6), pp.1309-1332.

[39] Montemerlo, M., Thrun, S., Koller, D. and Wegbreit, B., 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. Aaai/iaai, 593598.

[40] Ros, G., Sappa, A., Ponsa, D. and Lopez, A.M., 2012, June. Visual slam for driverless cars: A brief survey. In Intelligent Vehicles Symposium (IV) Workshops (Vol. 2).

[41] Liu, S., Tang, J., Wang, C., Wang, Q. and Gaudiot, J.L., 2017. A Unified Cloud Platform for Autonomous Driving. Computer, 50(12), pp.42-49.

[42] https://www.figure-eight.com/resources/human-in-the-loop/

[43] https://www.computerworld.com/article/3004013/robotics/why-human-in-the-loop-computing-is-the-future-of-machine-learning.html

[44] Xin, D., Ma, L., Liu, J., Macke, S., Song, S. and Parameswaran, A., 2018. Accelerating Human-in-the-loop Machine Learning: Challenges and Opportunities. arXiv preprint arXiv:1804.05892.

[45] Stanojevic, R., Abbar, S., Thirumuruganathan, S., Morales, G.D.F., Chawla, S., Filali, F. and Aleimat, A., 2018, January. Road Network Fusion for Incremental Map Updates. In LBS 2018: 14th International Conference on Location Based Services (pp. 91-109). Springer, Cham.

[46] https://blog.mapillary.com/update/2017/10/12/human-in-the-loop.html

[47] Chen, C., Lu, C., Huang, Q., Yang, Q., Gunopulos, D. and Guibas, L., 2016, August. City-scale map creation and updating using GPS collections. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1465-1474). ACM.

[48] https://www.mapillary.com/

[49] Dabeer, O., Gowaiker, R., Grzechnik, S.K., Lakshman, M.J., Reitmayr, G., Somasundaram, K., Sukhavasi, R.T. and Wu, X., 2017. An End-to-End System for Crowdsourced 3d Maps for Autonomous Vehicles: The Mapping Component. arXiv preprint arXiv:1703.10193.

[50] De Silva, V., Roche, J. and Kondoz, A., 2017. Fusion of LiDAR and Camera Sensor Data for Environment Sensing in Driverless Vehicles. arXiv preprint arXiv:1710.06230.

[51] Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F. and Herrtwich, R.G., 2013, December. Making bertha see. In Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on (pp. 214-221). IEEE.

[52] Massow, K., Kwella, B., Pfeifer, N., Häusler, F., Pontow, J., Radusch, I., Hipp, J., Dölitzscher, F. and Haueis, M., 2016, November. Deriving HD maps for highly automated driving from vehicular probe data. In Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on (pp. 1745-1752). IEEE.

[53] https://en.wikipedia.org/wiki/Autonomous_car#Testing

[54] https://techcrunch.com/2017/11/07/waymo-now-testing-its-self-driving-cars-on-public-roads-with-no-one-at-the-wheel/

[55] https://youtu.be/YXylqtEQ0tk?t=266

[56] https://youtu.be/Uj-rK8V-rik?t=640

[57] Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C.G. and Kaus, E., 2014. Making Bertha drive—An autonomous journey on a historic route. IEEE Intelligent Transportation Systems Magazine, 6(2), pp.8-20.

[58] https://lvl5.ai/

[59] Schreiber, M., Knöppel, C. and Franke, U., 2013, June. Laneloc: Lane marking based localization using highly accurate maps. In Intelligent Vehicles Symposium (IV), 2013 IEEE (pp. 449-454). IEEE.

[60] Levinson, J. and Thrun, S., 2010, May. Robust vehicle localization in urban environments using probabilistic maps. In Robotics and Automation (ICRA), 2010 IEEE International Conference on (pp. 4372-4378). IEEE.

[61] Biagioni, J. and Eriksson, J., 2012. Inferring road maps from global positioning system traces: Survey and comparative evaluation. Transportation Research Record: Journal of the Transportation Research Board, (2291), pp.61-71.

[62] Liu, X., Biagioni, J., Eriksson, J., Wang, Y., Forman, G. and Zhu, Y., Mining large-scale, sparse GPS traces for map inference. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD (Vol. 12, p. 669)

[63] https://techcrunch.com/2018/01/08/intels-mobileye-will-have-2-million-cars-on-roads-building-hd-maps-in-2018/

[64] https://www.wired.com/2014/12/google-maps-ground-truth/