2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Exploring OpenStreetMap Capability for Road Perception

Yang Zheng and Izzat H. Izzat

Abstract— **Drivable space detection or road perception is one of the most important tasks for autonomous driving. Sensor-based vision/laser systems may have limited performance in bad illumination/weather conditions, a prior knowledge of the road from the map data is expected to improve the effectiveness. This paper is to employ the map information extracted from the OpenStreetMap (OSM) data, and explore its capability for road perception. The OSM data can be used to render virtual street views, and further refined to provide the prior road mask. The OSM masks can be also combine with image processing and Lidar point clouding approaches to characterize the drivable space. Using a Fully Convolutional Neural Network (FCNN), the OSM availability for deep learning methods is also discussed.**

I. Introduction

Driving is a comprehensive task that incorporates the information among the driver, the vehicle, and the environment. Drivers make their judgement based on the environment, and then perform executions to control the vehicle. With the achievement of artificial intelligence, vehicle technologies have advanced significantly from the human driving towards fully automated driving. During the transition, the intelligent vehicle should be able to understand the driver's perception of the environment and controlling behavior of the vehicle, as well as provide human-liked interaction with the driver.

For autonomous driving perception, the effectiveness of sensor-based vision/laser systems is sometimes limited by the illumination and whether conditions (e.g. shadow, snow, etc.). A prior knowledge of the surrounding scenario is expected to assist these systems and contributes to understand the driving scenario. Nowadays, the high digital map database (e.g. Google Map, HERE Map, OpenStreetMap, etc.) is available to provide rich information about the static road, building and traffic infrastructure in the environment. Many researches have considered using the map data to fuse with other signals for the autonomous driving perception, vehicle localization, and road detection. Cao et al. [1] combines map data and camera image, improving the GPS positioning accuracy to localize the vehicle at the lane-level. Wang et al. [2] integrates the map data with image and Lidar signals, to provide a holistic 3D scene understanding on a single image, like object detection, vehicle pose estimation, semantic segmentation, etc. Alvarez et al. [3] considers using map data as a prior knowledge, and investigates multiple image-processing methods for the road detection. Seff et al. [4] retrieves road attributes from the map data, hence providing a visual sense description. Laddha et al. [5] uses the map data as the supervised annotation, and designs deep learning approaches for the road detection.

Yang Zheng and Izzat H. Izzat are with the Advanced Perception Center of Aptiv (previous Delphi Corporation) – Electronics & Safety, Agoura Hills, CA 91301 USA. (e-mail: {yang.zheng2, izzat.izzat}@aptiv.com).

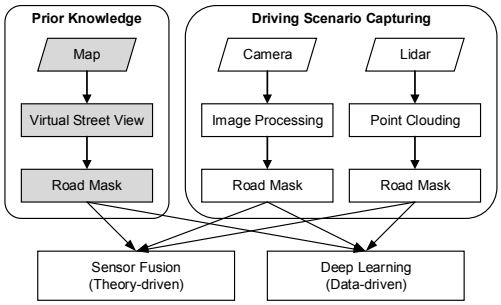

Figure 1. Overall framework to fuse OSM data for road perception.

The term of drivable space [6] is characterized as the static road surface restricted by the moving object occupations, where the static geo-location and rough shape of roads can be retrieved from the map data. This study is to make use of the map data, and explore how it can be integrated with other sensors for the road perception. After a comparison of the available map data, we selected the OpenStreetMap (OSM) data [7]. The major reason for selection is, the OSM data is open source and free, which makes it widely used in research activities. However, the limit is that the data is provided by the user contribution, and therefore the accuracy is not guaranteed. A commercial used high-digits map data is expected to provide a similar capability with higher precision.

Fig. 1 demonstrates the overall framework that fuses the OSM data with camera and Lidar signals for road perception. The OSM data is employed to render virtual street views and prior road masks will be refined from the virtual street views. Other kinds of road masks can be obtained from image processing and Lidar point clouding approaches. These masks will be fused together to characterize the drivable road surface. The availability of using OSM data for deep learning will also be discussed. The image and Lidar processing methods are theory-driven, whereas the deep learning approach is data-driven. Therefore, the OSM capability will be explored in the major two aspects.

II. Dataset Description

A. OpenStreetMap (OSM) Data

The OSM data can be accessed via its website[1], and the data within an area of interest can be downloaded by specifying a bounding box in terms of latitudes and longitudes. The data is given in XML format and is structured using three basic entities – nodes, ways, and relations [8, 9].

- *Nodes* – Nodes are point-shaped geometric elements which are used to represent point-of-interest (POI) like traffic signs and intersections. A node is described

[1] www.openstreetmap.org

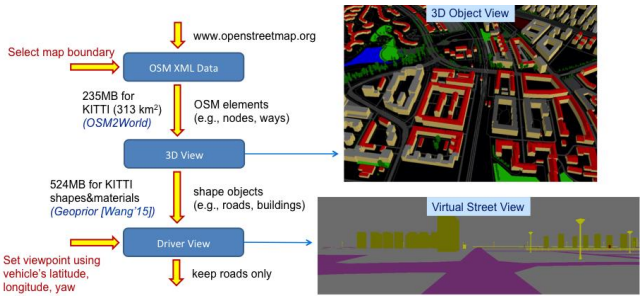

Figure 3. Rendering virtual street view from OSM, with approach flow diagram and output example.

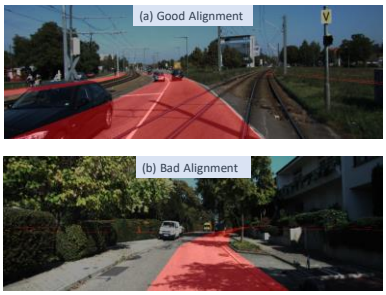

Figure 2. The effect of overlaying the OSM road mask (displayed in red color) onto the real camera image. (a) good alignment and (b) bad alignment.

by its GPS coordinates and a list of available tags. Each tag is formatted with a key and its corresponding value, describing the node attributes.

- *Ways* – Ways are used to model line-shaped or area-shaped geometric objects like road, railways, rivers, etc. A way entity is formatted as a collection of nodes.
- *Relations* – The relations are used to represent relationships between the aforementioned geometric entities to form more complicated structures.

B. KITTI Road Benchmark

For the road segmentation experiment, the KITTI Road dataset is introduced in this chapter [10]. KITTI is a designed computer vision benchmark suite for the real-world autonomous driving perception challenges. The KITTI dataset has been recorded from a moving platform (Volkswagen Passat station wagon) while driving in and around the mid-sized city of Karlsruhe, Germany. The testbed includes two color camera images on left and right side of the vehicle, one Velodyne 3D laser scanner, high-precision GPS measurements and IMU accelerations from a combined GPS/IMU system. The raw data has been further developed to provide stereo, optical flow, visual odometry, 3D object detection, 3D tracking, road detection, and semantic parsing sub-tasks. In this chapter, the road detection benchmark package has been utilized, which consists of 289 training and 290 test images. The images are grouped into three different categories of road scenes:

- *UU* – urban unmarked (98 training / 100 test)
- *UM* – urban marked (95 training / 96 test)
- *UMM* – urban multiple marked lanes (96 training / 94 test)

Ground truth for training images has been generated by manual annotations and is available for two different road terrain types: (i) road – the road area, (i.e, the composition of all lanes), and (2) lane – the ego-lane, (i.e., the lane the vehicle is currently driving on, only available for category "UM").

III. Virtual Street View Rendering

Fig. 2 demonstrates the general approach of virtual street view rendering, as well as two key output examples in the middle steps. As stated in Section II, the OSM data can be downloaded from its website by specifying a bounding box in terms of latitudes and longitudes. The data is structured in XML format, which describes node, way, and relation elements. The OSM2World toolkit [18] can be used to create 3D models from the OSM XML data. It renders a 3D virtual world from the bird-eye view, and generates a shape object (e.g. road, building, etc.) description file. When the vehicle's GPS coordinates and its heading direction is given, we can set it as a viewpoint and project the 3D virtual world into the driver's perspective [2]. If the interest to generate the road mask only, we can keep the road objects only and remove others, and make the virtual street view as a binary road mask.

The KITTI road dataset contains real images, Lidar point clouds, as well as GPS and IMU information, which makes it possible for us to compare the virtual street view against the real-world scenario. Fig.3 displays two examples that contain OSM road masks in the virtual street view overlaid on the real camera images. One advantage of the OSM road mask is, it can display the road surface area visually hidden by buildings or trees, and therefore provide more information on the road trajectory. There is a good alignment in Fig. 3-(a) and bad alignment in Fig. 3-(b). In the bad alignment, there is a shift between the virtual mask and the real road, which is caused by the GPS error; and a mismatch on the road width, which is caused by the OSM error. Since the OSM data is provided by the user contribution, the OSM accuracy cannot be guaranteed. Therefore, more processing steps are needed to reduce the error.

A. Super-pixel Refinement Approach

One label refinement approach is to look the image in super-pixels [5]. The processing steps are shown in Fig. 4. The super-pixel segmentation is based on the real camera image, using the K-means clustering method. Specific, an image size is 375*1242 pixels, and we select K=800 to generate 800 super-pixel segments. Initial pixel-wise labels from the OSM road mask are overlaid onto the real image and assigned to all the super-pixel segments. If a segment contains more than 50% initial road labels, it should be relabeled as a road super-pixel, and non-road otherwise. Once all the super-pixels are relabeled, the largest connected component will be selected as the refined road mask. It is worthwhile to note that this approach has been shown to reduce the boundary of the initial road mask; it brings in little new coverage and therefore may result in more false-negatives.

B. Multiple Candidates Approach

The super-pixel refinement output is still a binary road mask, an alternatively multiple candidates approach [1, 3] will create a confidence road mask. The processing steps are shown in Fig. 5. When the original vehicle GPS location and moving direction is projected onto the UTM flat plane, the

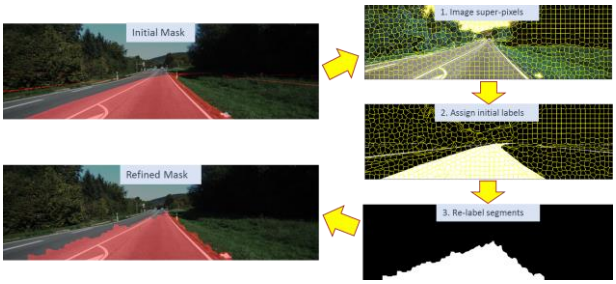

Figure 4. Super-pixel relabel approach for the road mask refinement, with processing steps and result example.

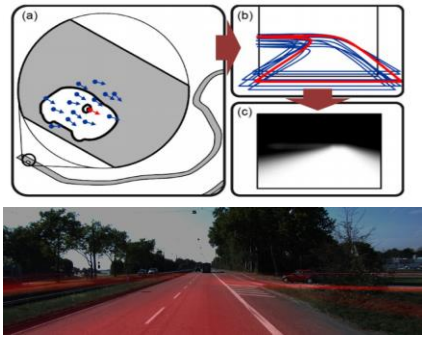

Figure 5. Multiple candidates approach to generate a confidence road mask, with processing steps and result example.

corresponding x, y coordinates and perspective angle is obtained. Considering the GPS error, a more precise position candidate could be located nearby, within the GPS variance range (e.g. x and y within ±1 meter, and angle within ±10 degrees). Therefore, N (e.g. N=100) possible viewpoint candidates are randomly selected within the variance range, based on a uniform or Gaussian distribution. The virtual street view will be rendered at the N viewpoint candidates, generating N binary road masks. The final confidence mask will be the overlay of all the candidate road masks, with an average on each pixel given by

$$P(x_i) = \frac{1}{N}\sum_{i=1}^{N} B_j(x_i) \qquad (1)$$

where x_i denotes the i-th pixel in the image, B_j denotes the j-th candidate binary mask, N denotes the number of candidates, and $P(x_i)$ denotes the probability for x_i being the road. As a contrary to the super-pixel refinement approach, the multiple candidate approach brings in more new coverage of the road area, and therefore will result in more false-positives.

IV. OSM for Sensor Fusion

Admittedly, the OSM data can only provide a prior knowledge of the static driving scenario, and the rendered virtual street view can only generate a coarse road mask. A more precise characterization of the drivable space still relies on vision/laser sensors. This sub-section will introduce two image processing methods and one Lidar point cloud processing method, and discuss how the OSM prior knowledge can be fused with these sensors.

A. GrabCut Algorithm

The GrabCut algorithm was designed by Rother et al. [11], which is an interactive approach to extract foreground and

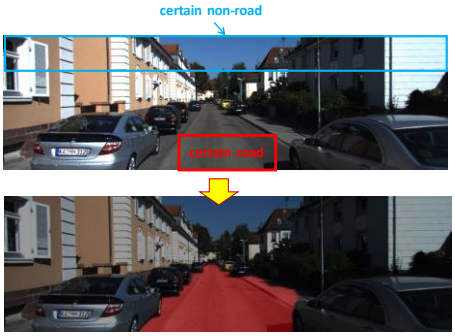

Figure 6. Example of GrabCut algorithm. Two hand-labeled rectangles are given on an image, and the algorithm outputs the road and non-road segments.

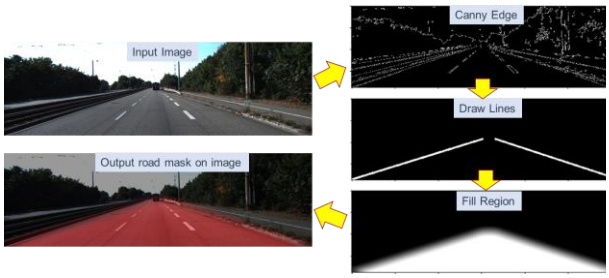

Figure 7. Lane-mark detection approach for road segmentation, with processing steps and result example.

background segmentations on an image. This approach requires a manually selected small area for the certain foreground region exclusively, and another small area for the certain background region. Based on these initial labels, two Gaussian Mixture Models are applied to model the foreground and background, and create the pixel distribution in terms of color statistics. All the other unlabeled pixels will be assigned with the probability of being the foreground or background. Next, a graph with a source node (connecting to the foreground) and a sink node (connecting to the background) is built from the pixel distribution, in which the other nodes represent the pixels and the weighted edges represent the probabilities. Then a min-cut algorithm is used to segment the graph, partitioning all foreground pixels connected to the source node and all background pixels connected to the sink node. The process is continued until classification converges or a pre-defined iteration step is reached. Fig. 6 shows an example how the GrabCut algorithm is utilized for the road detection. In our case, the top-bar is pre-selected as the certain non-road background, and the mid-bottom area is pre-selected as the certain road foreground. The GrabCut algorithm will start processing on these two rectangles, and segment the entire image into road and non-road regions. It is worthwhile to note that the GrabCut algorithm is based on the color, the effectiveness is limited if there is shadow or bad illumination conditions.

B. Lane-Mark Detection

Another widely used image processing method is to find out the lane-mark [12]. This approach is based on edges. Fig. 7 shows an example of the processing steps. First, a Canny edge detection is applied, according to the color gradients. Next, a Hough line transformation technique is utilized to draw lines with polynomial curve fitting. Finally, the region

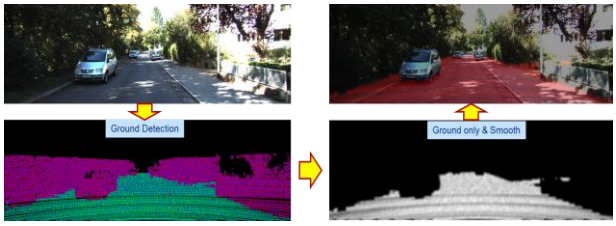

Figure 8. Lidar point cloud processing for ground detection, with processing steps and result example.

between the lane marks is filled as the road label. The advantage of the lane-mark detection approach is its robustness to image noises, however, it assumes that the lane marks are straight or almost straight, and therefore limit its effectiveness when there is a sharp-curved road or the lane boundaries are not clear.

C. Lidar Point Cloud Processing

Since the KITTI dataset provides Velodyne Lidar point clouds, Lidar-to-image calibration matrices and projection matrices, it is possible to project the points onto the image plane to view the points from the driver's perspective. Next, a multi-plane technique [13] is utilized to segment the ground plane against other buildings and objects. The points classified as ground are kept and others are removed. Due to the points resolution, they may look sparse on the image, and therefore a smoothing step is added to generate the filled road mask. An example of the processing steps is shown in Fig. 8. Since the ground detection is based on the height computation of the point-cloud planes, it is difficult to separate the road surface against the sidewalk or nearby grass.

D. Sensor Fusion

From the aforementioned approaches, five types of road mask are generated. Two masks from OSM (super-pixel refinement and multiple candidates), two from image processing (GrabCut algorithm and lane-mark approach), and one from Lidar point cloud processing. A combined road mask can be obtained by a weighted sum up of all these masks. Fig. 9 displays an example for the combined mask overlaid on the image, in which the weighting factor is chosen as 0.2 to consider all the five masks evenly. The weighting factors can be adjusted by how much confidence each approach is gained, or consider the weight selection as an optimization problem to calculate. If more masks were obtained by other approaches (e.g. depth layout), they can also be added into the combined mask.

Table I summarizes the road detection results using the masks generated from OSM, images, and Lidar. The experiments are examined using the three road types in KITTI – urban marked lane (UM), urban marked multi-lane (UMM), and urban unmarked lane (UU). The quantity evaluation is conducted on each pixel, and the results are measured using precision, recall and F1-score. Since the OSM multiple candidates approach and the combined output are represented as confidence masks, thresholds are selected to balance the precision-recall trade-off. Comparing the second and third row against the first row in Table I, it can be observed that the OSM super-pixel refinement approach increases the precision from the OSM direct rendering mask, and the OSM multiple candidates approach increases the recall. As we discussed

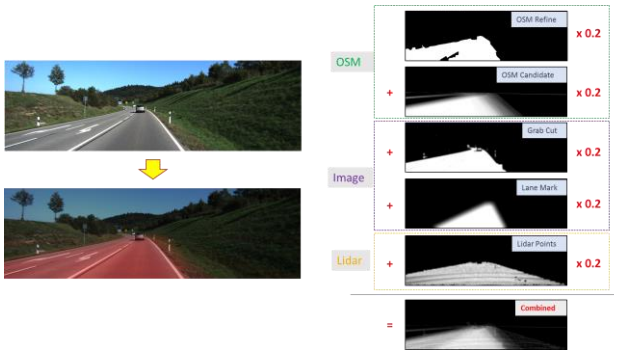

Figure 9. The combined road mask is obtained by a weighted sum up of the 5 aforementioned masks. The weighting factor in this example is selected 0.2, to consider the 5 masks evenly.

earlier, the super-pixel refinement approach removes out the boundary area while the multiple candidates approach brings in some new coverage. Although the Lidar point cloud processing achieves high recall values, its precision values are the lowest because the bicycle lane, sidewalk, and grass areas are included. The image processing GrabCut algorithm results in higher F1-score than other individual masks. The best result is obtained from the combined mask, in which a uniform weighting factor (i.e. 0.2) is applied on the five masks. Since the GrabCut algorithm provides the best individual result, it is believed to achieve higher combined mask result if a higher weight is selected on the GrabCut mask and lower weights on others.

V. OSM for Deep Learning

Recently, the deep learning approach has been proven successful in the computer vision area and for the autonomous driving applications. The deep learning approach does not require hand-crafted features, which is able to carry more information than the image processing approach. The road prior knowledge obtained from OSM is an additional source isolated from the image itself, and therefore it is interest to see how OSM can contribute to the road detection task using the deep learning approach.

The road detection is a semantic segmentation task, which can be addressed using a Fully Convolutional Neural Network (FCNN) [14]. Fig. 10 illustrates the FCNN architecture, which is composed of a VGG-16 convolutional network in the lower layers for down-sampling and classification, and an up-sampling layer on the top to recover the image to its original size. Overall it is an end-to-end structure, jointly learning to capture the semantics at its locations in the image.

To integrate the OSM prior information into the FCNN architecture, one typical approach is to use the rendered virtual road mask as an isolated image, or as an additional channel combining with an RGB color image. In summary, this study compares five different types of input images,

- *Camera image only* – color image given by KITTI.
- *OSM mask only* – using the confidence mask obtained by OSM multiple candidates approach.
- *Image + OSM* – adding the OSM mask as an addition channel to the color image.

TABLE I. ROAD DETECTION RESULTS USING OSM, IMAGES AND LIDAR MASKS

	urban marked lane (UM)			urban marked multi-lanes (UMM)			urban unmarked lane (UU)		
	Precision	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*
OSM direct	0.5123	0.8738	0.6177	0.7404	0.8023	0.7575	0.6689	0.6282	0.6319
OSM refinement	**0.5684**	0.8662	0.6555	**0.7974**	0.7713	0.7675	**0.7054**	0.6068	0.6368
OSM candidate	0.5516	**0.8833**	0.6441	0.7795	**0.8258**	0.7925	0.7815	**0.6598**	0.6952
Image GrabCut	0.6170	0.9300	**0.7135**	0.8724	0.8346	**0.8369**	0.7644	0.8600	**0.7889**
Image LaneMark	**0.6032**	0.8476	0.6722	**0.8475**	0.6167	0.7006	0.6602	**0.8096**	0.7114
Lidar PointCloud	**0.3334**	0.9885	0.4840	**0.5959**	0.9705	0.7340	**0.4670**	0.9113	0.5957
Combined	**0.6396**	**0.9253**	**0.7293**	**0.8962**	**0.8207**	**0.8504**	**0.8193**	**0.8169**	**0.8089**

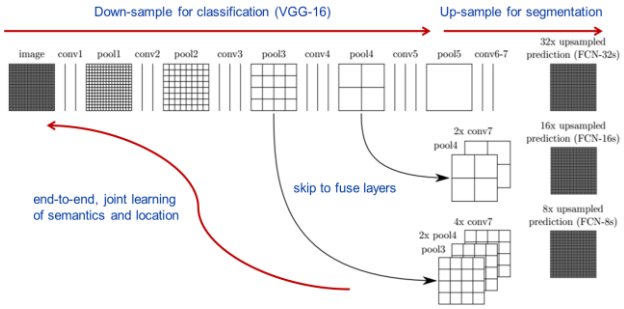

Figure 10. FCNN architecture for image semantic segmentation.

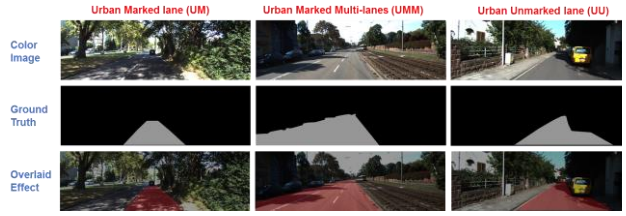

Figure 11. Comparison for different lane type scenarios. The lane type categories are grouped in three columns – UM, UMM, and UU. Color images, ground truths, and their overlaid effects are shown in rows.

- *Combined mask only* – using the overlaid mask obtained by combining OSM, image processing, and Lidar processing approaches.
- *Image + Combined* – adding the combined mask as an additional channel to the color image.

It is worthwhile to note the ground truth given for the supervised learning. KITTI provides 289 annotated images, it is not a big size but the images are categorized into three scenarios. As shown in Fig. 11, the UM ground truth is given only to cover the current ego lane, whereas the UMM and UU ground truth cover all the road surface. This will provide different supervision for the model to learn. However, due to the limited data size, we decide not to distinguish the categories in the training phase, but compare the difference on the test results.

Table II shows the road detection results using the deep learning approach, comparing variant inputs for the same FCNN model structure. The highest values in each column are highlighted, which are always higher than the image processing results shown in Table I. The most obvious finding is, the UM result is always lower than UM and UU, and its higher recall but lower precision. It is because the UM ground truth only covers the current ego lane, which is smaller than the predicted road surface. Depends on the task interest, we should decide how to supervise the model training. However,

Figure 12. Training loss decreasing history during the training iterations. Variant input types are compared in different colors.

the rendered OSM road masks cover all the road surface for the three categories. Even if the OSM node tags provide road attributes to decide the number of lanes, but without sufficient GPS accuracy, it is still challenging to locate the vehicle at the lane-level. This could be the limit of the current available OSM data.

In Table II, no matter what kind of inputs is given, image, mask, or any combination, the difference is not too much obvious. This may be possibly explained in Fig. 12, which compares the training loss history for the five input types. If sufficient training iteration is given, any of the five input types are finally converged. This is the advantage of deep learning. The only difference is their convergence speed. The "mask_only" input converges the fastest, because it is already composed of high-level features extracted from the image processing. For the "OSM_only" input, it does contain as much information as the color image, so it takes more time for the model to learn, but can still converge at last. This motivates several other possible availabilities to use the OSM data in deep learning for special cases.

- **Special Case (a): Use OSM virtual street views for test** – There may be several pre-trained models published for the road segmentation. Their models were trained using the camera street images and human annotation for the ground truth. It is expected to test their models in our experiments, but maybe the street view images are not always available in our testing area. Can we use the virtual street views rendered from OSM for the testing?
- **Special Case (b): Use OSM road mask for automatic annotation** – Sometimes a large dataset is available for us to train our own model and test, however the ground truth annotation is limited and requires massive human labor. Is it possible to trust the OSM and use its rendered road mask to provide the automatic annotation?

TABLE II. ROAD DETECTION RESULTS USING DEEP LEARNING

	urban marked lane (UM)			urban marked multi-lanes (UMM)			urban unmarked lane (UU)		
	Precision	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*	*Precision*	*Recall*	*F1-Score*
Image only	0.8101	0.9383	0.8492	0.9267	**0.8717**	**0.8858**	0.8490	0.9282	0.8768
OSM only	0.7984	0.9391	0.8418	0.9040	0.8571	0.8672	0.8538	0.9281	0.8800
Image + OSM	0.8044	0.9478	0.8475	0.9291	0.8605	0.8787	0.8361	0.9265	0.8680
Mask only	0.7665	**0.9492**	0.8251	0.9321	0.8454	0.8712	0.8620	0.9362	0.8896
Image + mask	**0.8281**	0.9387	**0.8648**	**0.9383**	0.8510	0.8750	**0.8664**	**0.9405**	**0.8922**

TABLE III. SPECIAL CASE RESULTS USING OSM IN DEEP LEARNING

(a) Train: camera; Annotation: human; Test: OSM

	Precision	*Recall*	*F1-score*
UM	0.4681	0.9647	0.6067
UMM	**0.8025**	**0.8373**	**0.8093**
UU	0.6191	0.8846	0.7157

(b) Train: camera; Annotation: OSM; Test: camera

	Precision	*Recall*	*F1-score*
UM	0.5536	0.8685	0.6477
UMM	**0.7932**	**0.7962**	**0.7803**
UU	0.7197	0.7080	0.6880

Table III lists the experimental results for the two special cases. It can be observed that the UMM performs better than the others. Again, it is because the OSM rendered road mask covers all the road surface not a single lane, which makes a bad performance for UM. UU roads are usually in the rural places with low traffic flows, which makes user contributions for OSM database limited in such areas. OSM cannot accurately estimate the unmarked lane width, and therefore performs bad for UU. This is the limit for OSM, but better results are expected if more accurate high digit map data were available.

VI. CONCLUSION

This paper demonstrated the capability of using OpenStreetMap data for road perception. The OSM data were used to render the virtual street view, which provided a road prior knowledge. We obtained five types of road mask – (1) OSM super-pixel refinement approach, (2) OSM multiple candidates approach, (3) color-based image GrabCut algorithm, (4) edge-based lane-mark approach, and (5) Lidar point cloud processing, and the weighted combination of them results the best performance. An FCNN architecture, which inputs the OSM mask as an additional channel, presented a better road segmentation than the hand-crafted masks. The OSM rendered virtual street views also have the capability to provide test images or the automatic annotation.

In addition to road segmentation on images, the OSM data also has the capability to describe the driving scenarios. Environmental attributes such as number of lanes, speed limit, distance to intersection, road curvature can be directly parsed from OSM XML format or computed in geometry. In the continued work, these attributes will be utilized with other signals for the traffic understanding such as motion planning and driving event recognition [15, 16, 17].

REFERENCES

[1] Cao, Gaoya, Florian Damerow, Benedict Flade, Markus Helmling, and Julian Eggert. 2016. "Camera to Map Alignment for Accurate Low-Cost Lane-Level Scene Interpretation."IEEE, . doi:10.1109/ITSC.2016.7795601.

[2] Wang, Shenlong, Sanja Fidler, and Raquel Urtasun. 2015. "Holistic 3D Scene Understanding from a Single Geo-Tagged Image."IEEE, . doi:10.1109/CVPR.2015.7299022.

[3] Alvarez, Jose M., Antonio M. Lopez, Theo Gevers, and Felipe Lumbreras. 2014. "Combining Priors, Appearance, and Context for Road Detection." IEEE Transactions on Intelligent Transportation Systems 15 (3): 1168-1178. doi:10.1109/TITS.2013.2295427.

[4] Seff, Ari and Jianxiong Xiao. 2016. "Learning from Maps: Visual Common Sense for Autonomous Driving." http://arxiv.org/abs/1611.08583.

[5] Laddha, Ankit, Mehmet Kemal Kocamaz, Luis E. Navarro-Serment, and Martial Hebert. 2016. "Map-Supervised Road Detection."IEEE, . doi:10.1109/IVS.2016.7535374.

[6] Moras, J., F. S. A. Rodriguez, V. Drevelle, G. Dherbomez, V. Cherfaoui, and P. Bonnifait. 2012. "Drivable Space Characterization using Automotive Lidar and Georeferenced Map Information."IEEE, . doi:10.1109/IVS.2012.6232252.

[7] Haklay, Mordechai and Patrick Weber. 2008. "Openstreetmap: User-Generated Street Maps." IEEE Pervasive Computing 7 (4): 12-18.

[8] Hentschel, Matthias and Bernardo Wagner. 2010. "Autonomous Robot Navigation Based on OpenStreetMap Geodata.". doi:10.1109/ITSC.2010.5625092.

[9] Floros, Georgios, Benito van der Zander, and Bastian Leibe. "Openstreetslam: Global vehicle localization using openstreetmaps." Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.

[10] Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. "Vision Meets Robotics: The KITTI Dataset." The International Journal of Robotics Research 32 (11): 1231-1237.

[11] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. Aug 1, 2004. "Grabcut: Interactive Foreground Extraction using Iterated Graph Cuts."ACM.

[12] Mariut, F., C. Fosalau, and D. Petrisor. 2012. "Lane Mark Detection using Hough Transform."IEEE, . doi:10.1109/ICEPE.2012.6463823.

[13] Himmelsbach, M., Felix v. Hundelshausen, and H. -J Wuensche. 2010. "Fast Segmentation of 3D Point Clouds for Ground Vehicles.". doi:10.1109/IVS.2010.5548059.

[14] Shelhamer, Evan, Jonathan Long, and Trevor Darrell. 2017. "Fully Convolutional Networks for Semantic Segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4): 640-651. doi:10.1109/TPAMI.2016.2572683.

[15] Zheng, Yang and John H. L. Hansen. 2017. "Lane-Change Detection from Steering Signal using Spectral Segmentation and Learning-Based Classification." IEEE Transactions on Intelligent Vehicles 2 (1): 14-24. doi:10.1109/TIV.2017.2708600.

[16] Hansen, John H.L., Carlos Busso, Yang Zheng, and Amardeep Sathyanarayana. "Driver Modeling for Detection and Assessment of Driver Distraction: Examples from the UTDrive Test Bed." IEEE Signal Processing Magazine 34, no. 4 (2017): 130-142.

[17] Zheng, Yang, Izzat H. Izzat and John H.L. Hansen, "Exploring OpenStreetMap Availability for Driving Environment Understanding", submitted for IEEE Transactions on Intelligent Vehicles, 2018.

[18] OSM2World toolkit, http://osm2world.org/