3D Fully Convolutional Network for Vehicle Detection in Point Cloud

Bo Li*

***Abstract*— 2D fully convolutional network has been recently successfully applied to object detection from images. In this paper, we extend the fully convolutional network based detection techniques to 3D and apply it to point cloud data. The proposed approach is verified on the task of vehicle detection from lidar point cloud for autonomous driving. Experiments on the KITTI dataset shows a significant performance improvement over the previous point cloud based detection approaches.**

I. INTRODUCTION

Understanding point cloud data has been recognized as an inevitable task for many robotic applications. Compared to image based detection, object detection in point cloud naturally localizes the 3D coordinates of the objects, which provides crucial information for subsequent tasks like navigation or manipulation.

In this paper, we design a 3D fully convolutional network (FCN) to detect and localize objects as 3D boxes from point cloud data. The 2D FCN [19] has achieved notable performance in image based detection tasks. The proposed approach extends FCN to 3D and is applied to 3D vehicle detection for an autonomous driving system, using a Velodyne 64E lidar. Meanwhile, the approach can be generalized to other object detection tasks on point cloud captured by Kinect, stereo or monocular structure from motion.

II. RELATED WORKS

A. 3D Object Detection in Point Cloud

A majority of 3D detection algorithms can be summarized as two stages, i.e. candidate proposal and classification. Candidates can be proposed by delicate segmentation algorithms [2], [5], [11], [14], [21], [22], [28], [29], [31], sliding window [30], random sampling [13], or the recently popular Region Proposal Network (RPN) [26]. For the classification stage, research have been drawn to features including shape model [6], [13] and geometry statistic features [22], [27], [29], [31]. Sparsing coding [4], [15] and deep learning [26] are also used for feature representation.

Besides directly operating in the 3D point cloud space, some other previous detection alogrithms project 3D point cloud onto 2D surface as depthmaps or range scans [3], [16], [17]. The projection inevitably loses or distorts useful 3D spatial information but can benefit from the well developed image based 2D detection algorithms.

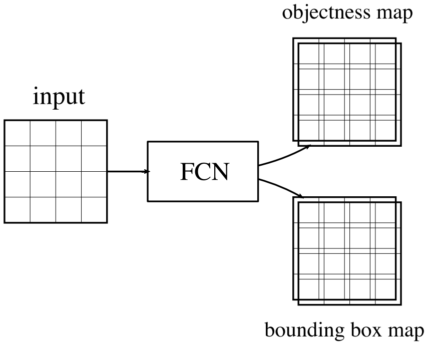

Fig. 1. A sample illustration of the structure of the FCN.

B. Convolutional Neural Network and 3D Object Detection

CNN based 3D object detection is recently drawing a growing attention in computer vision and robotics. [3], [9], [16], [17], [24], [25] embed 3D information in 2D projection and use 2D CNN for recognition or detection. [16] also suggest it possible to predict 3D object localization by 2D CNN network on range scans. [10] operates 3D voxel data but regards one dimension as a channel to apply 2D CNN. [8], [20], [26], [32] are among the very few earlier works on 3D CNN. [8], [20], [32] focus on object recognition and [26] proposes 3D R-CNN techniques for indoor object detection combining the Kinect image and point cloud.

In this paper, we transplant the fully convolutional network (FCN) to 3D to detect and localize object as 3D boxes in point cloud. The FCN is a recently popular framework for end-to-end object detection, with top performance in tasks including ImageNet, KITTI, ICDAR, etc. Variations of FCN include DenseBox [12], YOLO [23] and SSD [18]. The approach proposed in this paper is inspired by the basic idea of DenseBox.

III. APPROACH

A. FCN Based Detection Revisited

The procedure of FCN based detection frameworks can be summarized as two tasks, i.e. objectness prediction and bounding box prediction. As illustrated in Figure 1, a FCN is formed with two output maps corresponding to the two tasks respectively. The objectness map predicts if a region belongs to an object and the bounding box map predicts the coordinates of the object bounding box. We follow the denotion of [16]. Denote $\mathbf{o}_\mathbf{p}^a$ as the output at region $\mathbf{p}$ of the objectness map, which can be encoded by softmax or hinge

*Bo Li is a researcher at Baidu Inc. Contact: prclibo.github.io or libo24@baidu.com

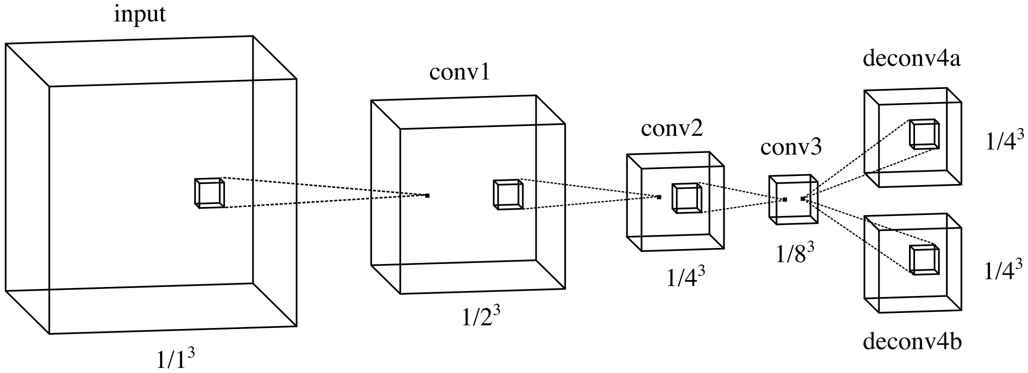

Fig. 2. A sample illustration of the 3D FCN structure used in this paper. Feature maps are first down-sampled by three convolution operation with the stride of $1/2^3$ and then up-samped by the deconvolution operation of the same stride. The output objectness map ($\mathbf{o}^a$) and bounding box map ($\mathbf{o}^b$) are collected from the deconv4a and deconv4b layers respectively.

loss. Denote $\mathbf{o}^b_\mathbf{p}$ as the output of the bounding box map, which is encoded by the coordinate offsets of the bounding box.

Denote the groundtruth objectness label at region $\mathbf{p}$ as $\ell_\mathbf{p}$. For simplicity each class corresponds to one label in this paper. In some works, e.g. SSD or DenseBox, the network can have multiple objectness labels for one class, corresponding to multiple scales or aspect ratios. The objectness loss at $\mathbf{p}$ is denoted as

$$\begin{aligned}\mathcal{L}_{\text{obj}}(\mathbf{p}) &= -\log(p_\mathbf{p}) \\ p_\mathbf{p} &= \frac{\exp(-\mathbf{o}^a_{\mathbf{p},\ell_\mathbf{p}})}{\sum_{\ell\in\{0,1\}}\exp(-\mathbf{o}^a_{\mathbf{p},\ell})}\end{aligned} \tag{1}$$

Denote the groundtruth bounding box coordinates offsets at region $\mathbf{p}$ as $\mathbf{b}_\mathbf{p}$. Similarly, in this paper we assume only one bounding box map is produced, though a more sophisticated network can have multiple bounding box offsets predicted for one class, corresponding to multiple scales or aspect ratios. Each bounding box loss is denoted as

$$\mathcal{L}_{\text{box}}(\mathbf{p}) = \|\mathbf{o}^b_\mathbf{p} - \mathbf{b}_\mathbf{p}\|^2 \tag{2}$$

The overall loss of the network is thus denoted as

$$\mathcal{L} = \sum_{\mathbf{p}\in\mathcal{P}} \mathcal{L}_{\text{obj}}(\mathbf{p}) + w \sum_{\mathbf{p}\in\mathcal{V}} \mathcal{L}_{\text{box}}(\mathbf{p}) \tag{3}$$

with w used to balance the objectness loss and the bounding box loss. $\mathcal{P}$ denotes all regions in the objectness map and $\mathcal{V} \in \mathcal{P}$ denotes all object regions. In the deployment phase, the regions with postive objectness prediction are selected. Then the bounding box predictions corresponding to these regions are collected and clustered as the detection results.

B. 3D FCN Detection Network for Point Cloud

Although a variety of discretization embedding have been introduced for high-dimensional convolution [1], [8], for simplicity we discretize the point cloud on square grids. The discretized data can be represented by a 4D array with dimensions of length, width, height and channels. For the simplest case, only one channel of value $\{0, 1\}$ is used to present whether there is any points observed at the corresponding grid elements. Some more sophisticated features have also been introduced in the previous works, e.g. [20].

The mechanism of 2D CNN naturally extends to 3D on the square grids. Figure 2 shows an example of the network structure used in this paper. The network follows and simplifies the hourglass shape from [19]. Layer conv1, conv2 and conv3 downsample the input map by $1/2^3$ sequentially. Layer deconv4a and deconv4b upsample the incoming map by 2^3 respectively. The ReLU activation is deployed after each layer. The output objectness map ($\mathbf{o}^a$) and bounding box map ($\mathbf{o}^b$) are collected from the deconv4a and deconv4b layers respectively.

Similar to DenseBox, the objectness region $\mathcal{V}$ is denoted as the center region of the object. For the proposed 3D case, a 3D sphere located at the object center is used. Points inside the sphere are labeled as positive / foreground label. The bounding box prediction at point $\mathbf{p}$ is encoded by the coordinate offsets, defined as:

$$\Delta\mathbf{b}_\mathbf{p} = (\mathbf{c}^\top_{\mathbf{p},1}, \mathbf{c}^\top_{\mathbf{p},2}, \ldots, \mathbf{c}^\top_{\mathbf{p},8})^\top - (\mathbf{p}^\top, \ldots, \mathbf{p}^\top) \tag{4}$$

where $\mathbf{c}_{\mathbf{p},\star}$ define the 3D coordinates of 8 corners of the object bounding box corresponding to the region $\mathbf{p}$.

The training and testing processes of the 3D CNN follows [16]. For the testing phase, candidate bounding boxes are extracted from regions predicted as objects and scored by counting its neighbors from all candidate bounding boxes. Bounding boxes are selected from the highest score and candidates overlapping with selected boxes are suppressed.

Figure 3 shows an example of the detection intermediate results. Bounding box predictions from objectness points are plotted as green boxes. Note that for severely occluded vehicles, the bounding boxes shape are distorted and not clustered. This is mainly due to the lack of similar samples in the training phase.

C. Comparison with 2D CNN

Compared to 2D CNN, the dimension increment of 3D CNN inevitably consumes more computational resource, mainly due to 1) the memory cost of 3D data embedding grids and 2) the increasing computation cost of convolving 3D kernels.

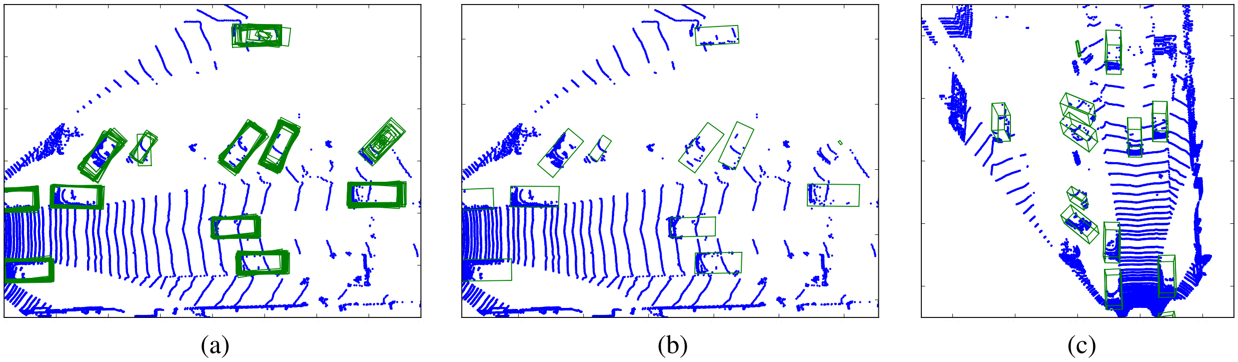

Fig. 3. Intermediate results of the 3D FCN detection procedure. (a) Bounding box predictions are collected from regions with high objectness confidence and are plotted as green boxes. (b) Bounding boxes after clustering plotted with the blue original point cloud. (c) Detection in 3D since (a) and (b) are visualized in the bird's eye view.

On the other hand, naturally embedding objects in 3D space avoids perspective distortion and scale variation in the 2D case. This make it possible to learn detection using a relatively simpler network structure.

IV. EXPERIMENTS

We evaluate the proposed 3D CNN on the vehicle detection task from the KITTI benchmark [7]. The task contains images aligned with point cloud and object info labeled by both 3D and 2D bounding boxes.

The experiments mainly focus on detection of the *Car* category for simplicity. Regions within the 3D center sphere of a *Car* are labeled as positive samples, i.e. in $\mathcal{V}$. *Van* and *Truck* are labeled to be ignored. *Pedestrian*, *Bicycle* and the rest of the environment are labeled as negative background, i.e. $\mathcal{P} - \mathcal{V}$.

The KITTI training dataset contains 7500+ frames of data, of which 6000 frames are randomly selected for training in the experiments. The rest 1500 frames are used for offline validation, which evaluates the detection bounding box by its overlap with groundtruth on the image plane and the ground plane. The detection results are also compared on the KITTI online evaluation, where only the image space overlap are evaluated.

The KITTI benchmark divides object samples into three difficulty levels. Though this is is originally designed for the image based detection, we find that these difficulty levels can also be approximately used in difficulty division for detection and evaluation in 3D. The minimum height of 40px for the easy level approximately corresponds to objects within 28m and the minimum height of 25px for the moderate and hard levels approximately corresponds to object within 47m.

A. Performance Analysis

The original KITTI benchmark assumes that detections are presented as 2D bounding boxes on the image plane. Then the overlap area of the image plane bounding box with its ground truth is measured to evaluate the detection. However, from the perspective of building a complete autonomous driving system, evaluation in the 2D image space does not well reflect the demand of the consecutive modules including planning and control, which usually operates in world space, e.g. in the full 3D space or on the ground plane. Therefore, in the offline evaluation, we validate the proposed approach in both the image space and the world space, using the following metrics:

- **Bounding box overlap on the image plane**. This is the original metric of the KITTI benchmark. The 3D bounding box detection is projected back to the image plane and the minimum rectangle hull of the projection is taken as the 2D bounding boxes. Some previous point cloud based detection methods [2], [16], [30] also use this metric for evaluation. A detection is accepted if the overlap area IoU with the groundtruth is larger than 0.7.
- **Bounding box overlap on the ground plane**. The 3D bounding box detection is projected onto the 2D ground plane orthogonally. A detection is accepted if the overlap area IoU with the groundtruth is larger than 0.7. This metric reflects the demand of the autonomous driving system naturally, in which the vertical localization of the vehicle is less important than the horizontal.

For the above metrics, the naive Average Precision (AP) and the Average Orientation Similarity (AOS) are both evaluated.

The performance of the proposed approach and [16] is listed in Table I. The proposed approach uses less layers and connections compared with [16] but achieves much better detection accuracy. This is mainly because objects have less scale variation and occlusion in 3D embedding. More detection results are visualized in Figure 4.

TABLE I
PERFORMANCE IN AVERAGE PRECISION AND AVERAGE ORIENTATION SIMILARITY FOR THE OFFLINE EVALUATION

		Easy	Moderate	Hard
Image Plane (AP)	Proposed	93.7%	81.9%	79.2%
	VeloFCN	74.1%	71.0%	70.0%
Image Plane (AOS)	Proposed	93.7%	81.8%	79.1%
	VeloFCN	73.9%	70.9%	69.9%
Ground Plane (AP)	Proposed	88.9%	77.3%	72.7%
	VeloFCN	77.3%	72.4%	69.4%
Ground Plane (AOS)	Proposed	88.9%	77.3%	72.7%
	VeloFCN	77.2%	72.3%	69.4%

TABLE II

Performance Comparison in Average Precision and Average Orientation Similarity for the KITTI Online Evaluation

		Easy	Moderate	Hard
Image Plane (AP)	Proposed	84.2%	75.3%	68.0%
	VeloFCN [16]	60.3%	47.5%	42.7%
	Vote3D [30]	56.8%	48.0%	42.6%
	CSoR	34.8%	26.1%	22.7%
	mBoW [2]	36.0%	23.8%	18.4%
Image Plane (AOS)	Proposed	84.1%	75.2%	67.9%
	VeloFCN [16]	59.1%	45.9%	41.1%
	CSoR	34.0%	25.4%	22.0%

B. KITTI Online Evaluation

The proposed approach is also evaluated on the KITTI online system. Note that on the current KITTI object detection benchmark image based detection algorithms outperforms previous point cloud based detection algorithms by a significant gap. This is due to two reasons: 1) The benchmark is using the metric of bounding box overlap on the image plane. Projecting 3D bounding boxes from point cloud inevitably introduce misalignment with 2D labeled bounding boxes. 2) Images have much higher resolution than point cloud (range scan), which enhances the detection of far or occluded objects.

The proposed approach is compared with previous point cloud based detection algorithms and the results are listed in Table II. The performance of our method outperforms previous methods by a significant gap of $> 20\%$, which is even comparable – though not as well as yet – with image based algorithms.

V. Conclusions

Recent study in deploying deep learning techniques in point cloud have shown the promising ability of 3D CNN to interpret shape features. This paper attempts to further push this research. To the best of our knowledge, this paper proposes the first 3D FCN framework for end-to-end 3D object detection. The performance improvement of this method is significant compared to previous point cloud based detection approaches. While in this paper the framework are experimented on the point cloud collected by Velodyne 64E under the scenario of autonomous driving, it naturally applies to point cloud created by other sensors or reconstruction algorithms.

ACKNOWLEDGMENT

The author would like to acknowledge the help from Xiaohui Li and Songze Li. Thanks also goes to Ji Wan and Tian Xia.

References

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, 29(2):753–762, 2010.

[2] Jens Behley, Volker Steinhage, and Armin B. Cremers. Laser-based segment classification using a mixture of bag-of-words. *IEEE International Conference on Intelligent Robots and Systems*, (1):4195–4200, 2013.

[3] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[4] Mark De Deuge, F Robotics, and Alastair Quadros. Unsupervised Feature Learning for Classification of Outdoor 3D Scans. *Araa.Asn.Au*, pages 2–4, 2013.

[5] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, a. Quadros, P. Morton, and a. Frenkel. On the segmentation of 3D lidar point clouds. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2798–2805, 2011.

[6] O.D. Faugeras and M. Hebert. The Representation, Recognition, and Locating of 3-D Objects. *The International Journal of Robotics Research*, 5(3):27–52, 1986.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[8] Ben Graham. Sparse 3D convolutional neural networks. *Bmvc*, pages 1–11, 2015.

[9] S Gupta, R Girshick, P Arbeláez, and J Malik. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. *arXiv preprint arXiv:1407.5736*, pages 1–16, 2014.

[10] Vishakh Hegde and Reza Zadeh. FusionNet: 3D Object Classification Using Multiple Data Representations.

[11] Michael Himmelsbach, Felix V Hundelshausen, and Hans-Joachim Wünsche. Fast segmentation of 3d point clouds for ground vehicles. *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 560–565, 2010.

[12] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. DenseBox: Unifying Landmark Localization with End to End Object Detection. pages 1–13, 2015.

[13] Andrew E Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[14] Klaas Klasing, Dirk Wollherr, and Martin Buss. A clustering method for efficient segmentation of 3D laser data. *Conference on Robotics and Automation, ICRA 2008. IEEE International*, pages 4043–4048, 2008.

[15] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised Feature Learning for 3D Scene Labeling. *IEEE International Conference on Robotics and Automation (ICRA 2014)*, pages 3050–3057, 2014.

[16] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *Proceedings of Robotics: Science and Systems*, 2016.

[17] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1417–1424, 2013.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed. SSD: Single Shot MultiBox Detector. *Arxiv*, 2015.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[20] Daniel Maturana and Sebastian Scherer. VoxNet : A 3D Convolutional Neural Network for Real-Time Object Recognition. pages 922–928, 2015.

[21] Frank Moosmann, Oliver Pink, and Christoph Stiller. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 215–220, 2009.

[22] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation - Supervoxels for point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2027–2034, 2013.

[23] Joseph Redmon, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *arXiv*, 2015.

[24] Max Schwarz, Hannes Schulz, and Sven Behnke. RGB-D Object Recognition and Pose Estimation based on Pre-trained Convolutional Neural Network Features. *IEEE International Conference on Robotics and Automation (ICRA)*, (May), 2015.

[25] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep learning for 3d object classification. *Advances in Neural Information Processing Systems*, pages 665–673, 2012.

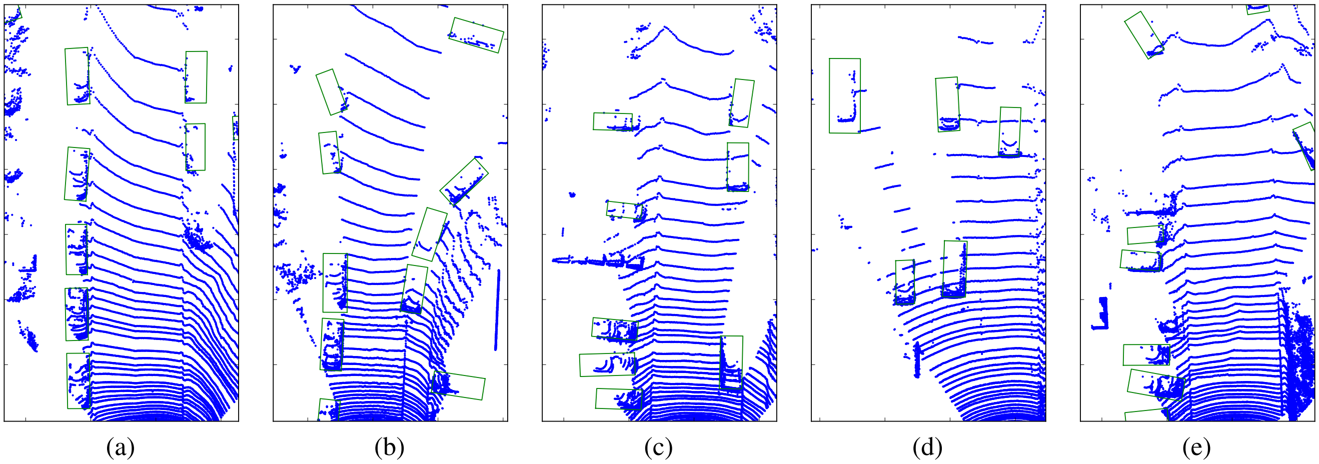

Fig. 4. More detection results on the KITTI dataset using 3D FCN.

[26] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. pages 634–651, 2014.

[27] Alex Teichman, Jesse Levinson, and Sebastian Thrun. Towards 3D object recognition via classification of arbitrary object tracks. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4034–4041, 2011.

[28] Rudolph Triebel, Richard Schmidt, Óscar Martínez Mozos, and Wolfram Burgard. Instance-based amn classification for improved object recognition in 2d and 3d laser range data. *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2225–2230, 2007.

[29] Rudolph Triebel, Jiwon Shin, and Roland Siegwart. Segmentation and Unsupervised Part-based Discovery of Repetitive Objects. *Robotics: Science and Systems*, 2006.

[30] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. *Proceedings of Robotics: Science and Systems, Rome, Italy*, 2015.

[31] Dominic Zeng Wang, Ingmar Posner, and Paul Newman. What could move? Finding cars, pedestrians and bicyclists in 3D laser data. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4038–4044, 2012.

[32] Zhirong Wu and Shuran Song. 3D ShapeNets : A Deep Representation for Volumetric Shapes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, pages 1–9, 2015.