Robotics: Science and Systems 2016
Ann Arbor, MI, USA, June 18-22, 2016

Vehicle Detection from 3D Lidar Using Fully Convolutional Network

Bo Li, Tianlei Zhang and Tian Xia
Baidu Research – Institute for Deep Learning
{libo24, zhangtianlei, xiatian}@baidu.com

***Abstract*—Convolutional network techniques have recently achieved great success in vision based detection tasks. This paper introduces the recent development of our research on transplanting the fully convolutional network technique to the detection tasks on 3D range scan data. Specifically, the scenario is set as the vehicle detection task from the range data of Velodyne 64E lidar. We proposes to present the data in a 2D point map and use a single 2D end-to-end fully convolutional network to predict the objectness confidence and the bounding boxes simultaneously. By carefully design the bounding box encoding, it is able to predict full 3D bounding boxes even using a 2D convolutional network. Experiments on the KITTI dataset shows the state-of-the-art performance of the proposed method.**

I. Introduction

For years of the development of robotics research, 3D lidars have been widely used on different kinds of robotic platforms. Typical 3D lidar data present the environment information by 3D point cloud organized in a range scan. A large number of research have been done on exploiting the range scan data in robotic tasks including localization, mapping, object detection and scene parsing [16].

In the task of object detection, range scans have an specific advantage over camera images in localizing the detected objects. Since range scans contain the spatial coordinates of the 3D point cloud by nature, it is easier to obtain the pose and shape of the detected objects. On a robotic system including both perception and control modules, e.g. an autonomous vehicle, accurately localizing the obstacle vehicles in the 3D coordinates is crucial for the subsequent planning and control stages.

In this paper, we design a fully convolutional network (FCN) to detect and localize objects as 3D boxes from range scan data. FCN has achieved notable performance in computer vision based detection tasks. This paper transplants FCN to the detection task on 3D range scans. We strict our scenario as 3D vehicle detection for an autonomous driving system, using a Velodyne 64E lidar. The approach can be generalized to other object detection tasks on other similar lidar devices.

II. Related Works

A. Object Detection from Range Scans

Tranditional object detection algorithms propose candidates in the point cloud and then classify them as objects. A common category of the algorithms propose candidates by segmenting the point cloud into clusters. In some early works, rule-based segmentation is suggested for specific scene [10, 20, 5]. For example when processing the point cloud captured by an autonomous vehicle, simply removing the ground plane and cluster the remaining points can generate reasonable segmentation [10, 5]. More delicate segmentation can be obtained by forming graphs on the point cloud [32, 14, 21, 29, 30]. The subsequent object detection is done by classifying each segments and thus is sometimes vulnerable to incorrect segmentation. To avoid this issue, Behley et al. [2] suggests to segment the scene hierarchically and keep segments of different scales. Other methods directly exhaust the range scan space to propose candidates to avoid incorrect segmentation. For example, Johnson and Hebert [13] randomly samples points from the point cloud as correspondences. Wang and Posner [31] scan the whole space by a sliding window to generate proposals.

To classify the candidate data, some early researches assume known shape model and match the model to the range scan data [6, 13]. In recent machine learning based detection works, a number of features have been hand-crafted to classify the candidates. Triebel et al. [29], Wang et al. [32], Teichman et al. [28] use shape spin images, shape factors and shape distributions. Teichman et al. [28] also encodes the object moving track information for classification. Papon et al. [21] uses FPFH. Other features include normal orientation, distribution histogram and etc. A comparison of features can be found in [1]. Besides the hand-crafted features, Deuge et al. [4], Lai et al. [15] explore to learn feature representation of point cloud via sparse coding.

We would also like to mention that object detection on RGBD images [3, 17] is closely related to the topic of object detection on range scan. The depth channel can be interpreted as a range scan and naturally applies to some detection algorithms designed for range scan. On the other hand, numerous researches have been done on exploiting both depth and RGB information in object detection tasks. We omit detailed introduction about traditional literatures on RGBD data here but the proposed algorithm in this paper can also be generalized to RGBD data.

B. Convolutional Neural Network on Object Detection

The Convolutional Neural Network (CNN) has achieved notable succuess in the areas of object classification and detection on images. We mention some state-of-the-art CNN based detection framework here. R-CNN [8] proposes candidate regions and uses CNN to verify candidates as valid objects.

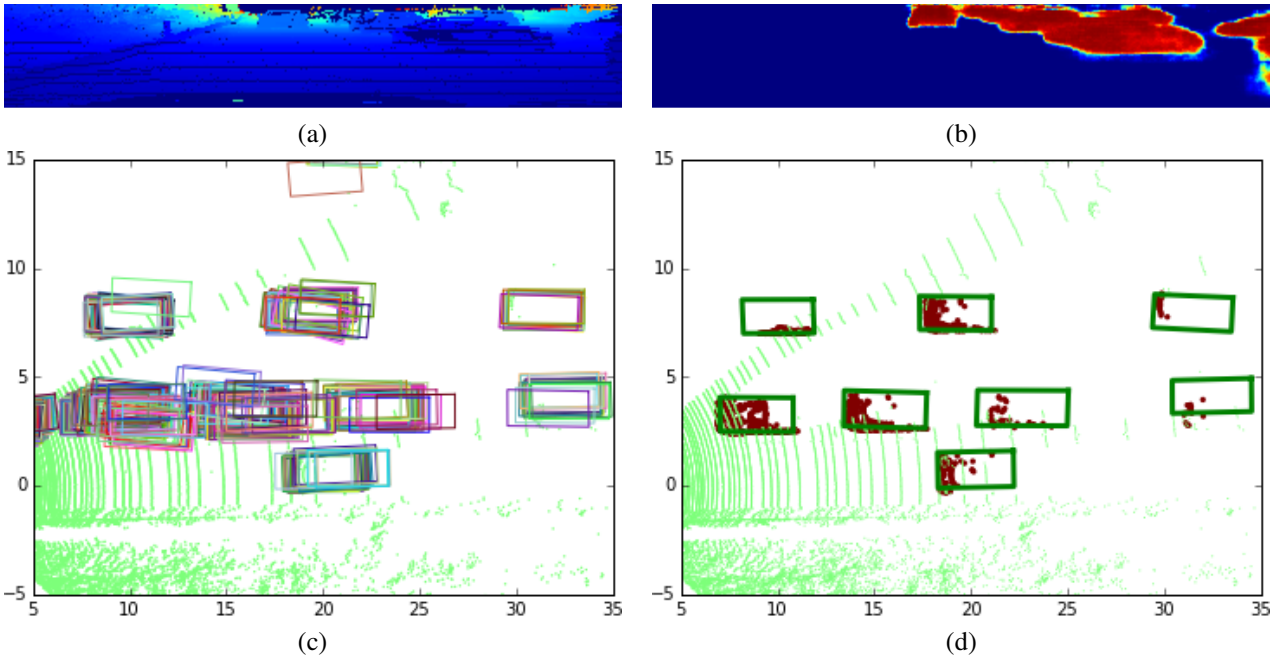


Fig. 1. Data visualization generated at different stages of the proposed approach. (a) The input point map, with the d channel visualized. (b) The output confidence map of the objectness branch at $\mathbf{o}^{o}_{\mathbf{p}}$. Red denotes for higher confidence. (c) Bounding box candidates corresponding to all points predicted as positive, i.e. high confidence points in (b). (d) Remaining bounding boxes after non-max suppression. Red points are the groundtruth points on vehicles for reference.

OverFeat [25], DenseBox [11] and YOLO [23] uses end-to-end unified FCN frameworks which predict the objectness confidence and the bounding boxes simultaneously over the whole image. Some research has also been focused on applying CNN on 3D data. For example on RGBD data, one common aspect is to treat the depthmaps as image channels and use 2D CNN for classification or detection [9, 24, 26]. For 3D range scan some works discretize point cloud along 3D grids and train 3D CNN structure for classification [33, 19]. These classifiers can be integrated with region proposal method like sliding window [27] for detection tasks. The 3D CNN preserves more 3D spatial information from the data than 2D CNN while 2D CNN is computationally more efficient.

In this paper, our approach project range scans as 2D maps similar to the depthmap of RGBD data. The frameworks of Huang et al. [11], Sermanet et al. [25] are transplanted to predict the objectness and the 3D object bounding boxes in a unified end-to-end manner.

III. Approach

A. Data Preparation

We consider the point cloud captured by the Velodyne 64E lidar. Like other range scan data, points from a Velodyne scan can be roughly projected and discretized into a 2D point map, using the following projection function.

$$\begin{aligned} \theta &= \text{atan2}(y, x) \\ \phi &= \arcsin(z/\sqrt{x^2+y^2+z^2}) \\ r &= \lfloor \theta/\Delta\theta \rfloor \\ c &= \lfloor \phi/\Delta\phi \rfloor \end{aligned} \tag{1}$$

where $\mathbf{p} = (x, y, z)^\top$ denotes a 3D point and (r, c) denotes the 2D map position of its projection. θ and ϕ denote the azimuth and elevation angle when observing the point. $\Delta\theta$ and $\Delta\phi$ is the average horizontal and vertical angle resolution between consecutive beam emitters, respectively. The projected point map is analogous to cylindral images. We fill the element at (r, c) in the 2D point map with 2-channel data (d, z) where $d = \sqrt{x^2+y^2}$. Note that x and y are coupled as d for rotation invariance around z. An example of the d channel of the 2D point map is shown in Figure 1a. Rarely some points might be projected into a same 2D position, in which case the point nearer to the observer is kept. Elements in 2D positions where no 3D points are projected into are filled with $(d, z) = (0, 0)$.

B. Network Architecture

The trunk part of the proposed CNN architecture is similar to Huang et al. [11], Long et al. [18]. As illustrated in Figure 2, the CNN feature map is down-sampled consecutively in the first 3 convolutional layers and up-sampled consecutively in deconvolutional layers. Then the trunk splits at the 4th layer into a objectness classification branch and a 3D bounding box regression branch. We describe its implementation details as follows:

- The input point map, output objectness map and bounding box map are of the same width and height, to provide point-wise prediction. Each element of the objectness map predicts whether its corresponding point is on a vehicle. If the corresponding point is on a vehicle, its corresponding element in the bounding box map predicts the 3D bounding box of the belonging vehicle. Section

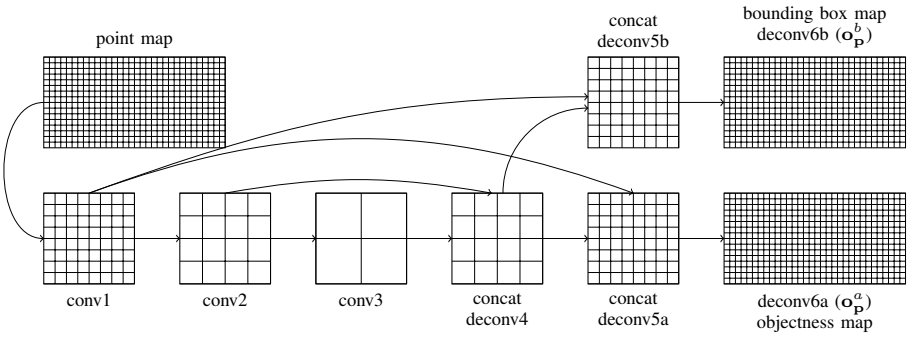


Fig. 2. The proposed FCN structure to predict vehicle objectness and bounding box simultaneously. The output feature map of conv1/deconv5a, conv1/deconv5b and conv2/deconv4 are first concatenated and then ported to their consecutive layers, respectively.

III-C explains how the objectness and bounding box is encoded.

- In conv1, the point map is down-sampled by 4 horizontally and 2 vertically. This is because for a point map captured by Velodyne 64E, we have approximately $\Delta\phi = 2\Delta\theta$, i.e. points are denser on horizotal direction. Similarly, the feature map is up-sampled by this factor of (4, 2) in deconv6a and deconv6b, respectively. The rest conv/deconv layers all have equal horizontal and vertical resolution, respectively, and use squared strides of $(2, 2)$ when up-sampling or down-sampling.
- The output feature map pairs of conv3/deconv4, conv2/deconv5a, conv2/deconv5b are of the same sizes, respectively. We concatenate these output feature map pairs before passing them to the subsequent layers. This follows the idea of Long et al. [18]. Combining features from lower layers and higher layers improves the prediction of small objects and object edges.

C. Prediction Encoding

We now describe how the output feature maps are defined. The objectness map deconv6a consists of 2 channels corresponding to foreground, i.e. the point is on a vehicle, and background. The 2 channels are normalized by softmax to denote the confidence.

The encoding of the bounding box map requires some extra conversion. Consider a lidar point $\mathbf{p} = (x, y, z)$ on a vehicle. Its observation angle is (θ, ϕ) by (1). We first denote a rotation matrix $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{R}_z(\theta)\mathbf{R}_y(\phi) \tag{2}$$

where $\mathbf{R}_z(\theta)$ and $\mathbf{R}_y(\phi)$ denotes rotations around z and y axes respectively. If denote $\mathbf{R}$ as $(\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z)$, $\mathbf{r}_x$ is of the same direction as $\mathbf{p}$ and $\mathbf{r}_y$ is parallel with the horizontal plane. Figure 3a illustrate an example on how $\mathbf{R}$ is formed. A bounding box corner $\mathbf{c}_\mathbf{p} = (x_c, y_c, z_c)$ is thus transformed as:

$$\mathbf{c}'_\mathbf{p} = \mathbf{R}^\top(\mathbf{c}_\mathbf{p} - \mathbf{p}) \tag{3}$$

Our proposed approach uses $\mathbf{c}'_\mathbf{p}$ to encode the bounding box corner of the vehicle which $\mathbf{p}$ belongs to. The full bounding box is thus encoded by concatenating 8 corners in a 24d vector as

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

Corresponding to this 24d vector, deconv6b outputs a 24-channel feature map accordingly.

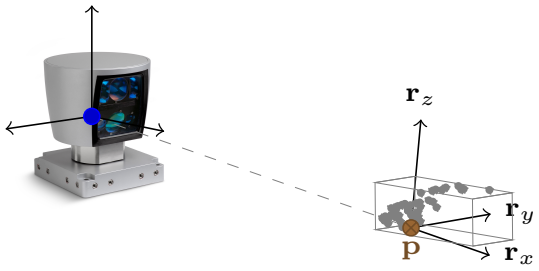
The transform (3) is designed due to the following two reasons:

- *Translation part* Compared to $\mathbf{c}_\mathbf{p}$ which distributes over the whole lidar perception range, e.g. $[-100\text{m}, 100\text{m}] \times [-100\text{m}, 100\text{m}]$ for Velodyne, the corner offset $\mathbf{c}_\mathbf{p} - \mathbf{p}$ distributes in a much smaller range, e.g. within size of a vehicle. Experiments show that it is easier for the CNN to learn the latter case.
- *Rotation part* $\mathbf{R}^\top$ ensures the rotation invariance of the corner coordinate encoding. When a vehicle is moving around a circle and one observes it from the center, the appearance of the vehicle does not change in the observed range scan but the bounding box coordinates vary in the range scan coordinate system. Since we would like to ensure that same appearances result in same bounding box prediction encoding, the bounding box coordinates are rotated by $\mathbf{R}^\top$ to be invariant. Figure 3b illustrates a simple case. Vehicle A and B have the same appearance for an observer at the center, i.e. the right side is observed. Vehicle C has a difference appearance, i.e. the rear-right part is observed. With the conversion of (3), the bounding box encoding $\mathbf{b}'_\mathbf{p}$ of A and B are the same but that of C is different.

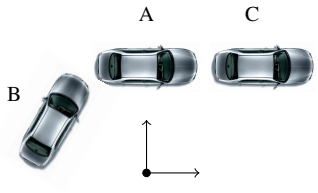
D. Training Phase

1) Data Augmentation: Similar to the training phase of a CNN for images, data augmentation significantly enhances the network performance. For the case of images, training data are usually augmented by randomly zooming or rotating the original images to synthesis more training samples. For the case of range scans, simply applying these operations results in variable $\Delta\theta$ and $\Delta\phi$ in (1), which violates the geometry property of the lidar device. To synthesis geometrically correct 3D range scans, we randomly generate a 3D transform near identity. Before projecting point cloud by (1), the random transform is applied the point cloud. The translation component of the transform results in zooming effect of the synthesized range scan. The rotation component results in rotation effect of the range scan.

2) Multi-Task Training: As illustrated Section III-B, the proposed network consists of one objectness classification



(a)



(b)

Fig. 3. (a) Illustration of (3). For each vehicle point $\mathbf{p}$, we define a specific coordinate system which is centered at $\mathbf{p}$. The x axis ($\mathbf{r}_x$) of the coordinate system is along with the ray from Velodyne origin to $\mathbf{p}$ (dashed line). (b) An example illustration about the rotation invariance when observing a vehicle. Vehicle A and B have same appearance. See (3) in Section III-C for details.

branch and one bounding box regression branch. We respectively denote the losses of the two branches in the training phase. As notation, denote $\mathbf{o}^a_\mathbf{p}$ and $\mathbf{o}^b_\mathbf{p}$ as the feature map output of deconv6a and deconv6b corresponding to point $\mathbf{p}$ respectively. Also denote $\mathcal{P}$ as the point cloud and $\mathcal{V} \subset \mathcal{P}$ as all points on all vehicles.

The loss of the objectness classification branch corresponding to a point $\mathbf{p}$ is denoted as a softmax loss

$$\begin{aligned} \mathcal{L}_{\text{obj}}(p) &= -\log(p_\mathbf{p}) \\ p_\mathbf{p} &= \frac{\exp(-\mathbf{o}^a_{\mathbf{p},l_\mathbf{p}})}{\sum_{l\in\{0,1\}}\exp(-\mathbf{o}^a_{\mathbf{p},l})} \end{aligned} \tag{5}$$

where $l_\mathbf{p} \in \{0,1\}$ denotes the groundtruth objectness label of $\mathbf{p}$, i.e. 0 as background and 1 as a point on vechicles. $o^a_{\mathbf{p},\star}$ denotes the deconv6a feature map output of channel $\star$ for point $\mathbf{p}$.

The loss of the bounding box regression branch corresponding to a point $\mathbf{p}$ is denoted as a L2-norm loss

$$\mathcal{L}_{\text{box}}(p) = \|\mathbf{o}^b_\mathbf{p} - \mathbf{b}'_\mathbf{p}\|^2 \tag{6}$$

where $\mathbf{b}'_\mathbf{p}$ is a 24d vector denoted in (4). Note that $\mathcal{L}_{\text{box}}$ is only computed for those points on vehicles. For non-vehicle points, the bounding box loss is omitted.

3) Training strategies: Compared to positive points on vehicles, negative (background) points account for the majority portion of the point cloud. Thus if simply pass all objectness losses in (5) in the backward procedure, the network prediction will significantly bias towards negative samples. To avoid this effect, losses of positive and negative points need to be balanced. Similar balance strategies can be found in Huang et al. [11] by randomly discarding redundant negative losses. In our training procedure, the balance is done by keeping all negative losses but re-weighting them using

$$w_1(\mathbf{p}) = \begin{cases} k|\mathcal{V}|/(|\mathcal{P}|-|\mathcal{V}|) & \mathbf{p}\in\mathcal{P}-\mathcal{V} \\ 1 & \mathbf{p}\in\mathcal{V} \end{cases} \tag{7}$$

which denotes that the re-weighted negative losses are averagely equivalent to losses of $k|\mathcal{V}|$ negative samples. In our case we choose $k = 4$. Compared to randomly discarding samples, the proposed balance strategy keeps more information of negative samples.

Additionally, near vehicles usually account for larger portion of points than far vehicles and occluded vehicles. Thus vehicle samples at different distances also need to be balanced. This helps avoid the prediction to bias towards near vehicles and neglect far vehicles or occluded vehicles. Denote $n(\mathbf{p})$ as the number of points belonging to the same vehicle with $\mathbf{p}$. Since the 3D range scan points are almost uniquely projected onto the point map. $n(\mathbf{p})$ is also the area of the vehicle of $\mathbf{p}$ on the point map. Denote $\bar{n}$ as the average number of points of vehicles in the whole dataset. We re-weight $\mathcal{L}_{\text{obj}}(\mathbf{p})$ and $\mathcal{L}_{\text{box}}(\mathbf{p})$ by w_2 as

$$w_2(\mathbf{p}) = \begin{cases} \bar{n}/n(\mathbf{p}) & \mathbf{p}\in\mathcal{V} \\ 1 & \mathbf{p}\in\mathcal{P}-\mathcal{V} \end{cases} \tag{8}$$

Using the losses and weights designed above, we accumulate losses over deconv6a and deconv6b for the final training loss

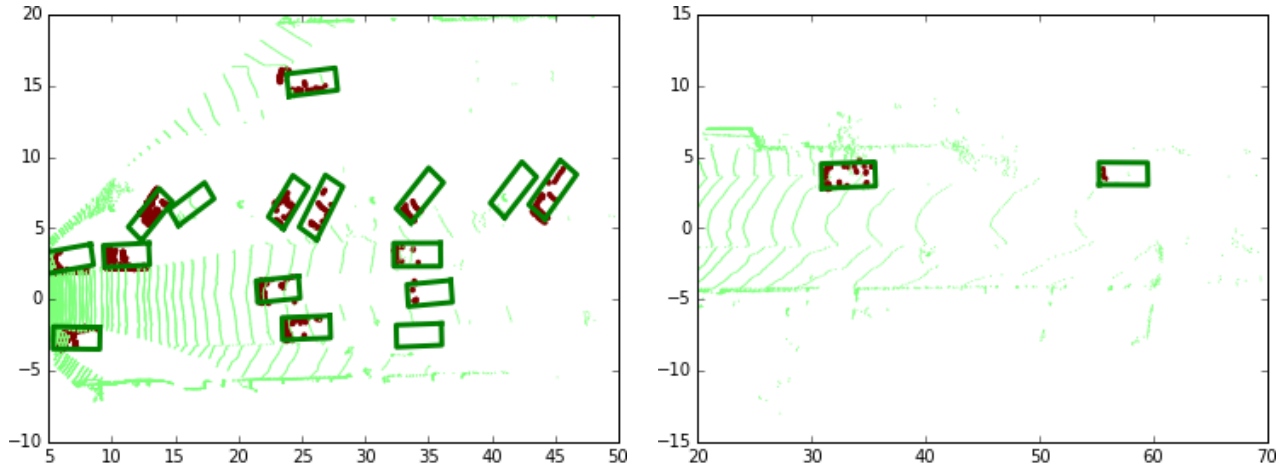
$$\mathcal{L} = \sum_{\mathbf{p}\in\mathcal{P}} w_1(\mathbf{p})w_2(\mathbf{p})\mathcal{L}_{\text{obj}}(\mathbf{p}) + w_{\text{box}}\sum_{\mathbf{p}\in\mathcal{V}} w_2(\mathbf{p})\mathcal{L}_{\text{box}}(\mathbf{p}) \tag{9}$$

with w_{box} used to balance the objectness loss and the bounding box loss.

E. Testing Phase

During the test phase, a range scan data is fed to the network to produce the objectness map and the bounding box map. For each point which is predicted as positive in the objectness map, the corresponding output $\mathbf{o}^b_\mathbf{p}$ of the bounding box map is splitted as $\mathbf{c}'_{\mathbf{p},i}, i = 1,\ldots,8$. $\mathbf{c}'_{\mathbf{p},i}$ is then converted to box corner $\mathbf{c}_{\mathbf{p},i}$ by the inverse transform of (3). We denote each bounding box candidates as a 24d vector $\mathbf{b}_\mathbf{p} = (\mathbf{c}^\top_{\mathbf{p},1}, \mathbf{c}^\top_{\mathbf{p},2}, \cdots, \mathbf{c}^\top_{\mathbf{p},8})^\top$. The set of all bounding box candidates is denoted as $\mathbf{B} = \{\mathbf{b}_\mathbf{p} | \mathbf{o}^a_{\mathbf{p},1} > \mathbf{o}^a_{\mathbf{p},0}\}$. Figure 1c shows the bounding box candidates of all the points predicted as positive.

We next cluster the bounding boxes and prune outliers by a non-max suppression strategy. Each bounding box $\mathbf{b}_\mathbf{p}$ is scored by counting its neighbor bounding boxes in $\mathbf{B}$ within a distance δ, denoted as $\#\{\mathbf{x}\in\mathbf{B} | \|\mathbf{x}-\mathbf{b}_\mathbf{p}\| < \delta\}$. Bounding boxes are picked from high score to low score. After one box is picked, we find out all points inside the bounding box and remove their corresponding bounding box candidates from $\mathbf{B}$. Bounding box candidates whose score is lower than 5 is discarded as outliers. Figure 1d shows the picked bounding boxes for Figure 1a.



(a) (b)

Fig. 4. More examples of the detection results. See Section IV-A for details. (a) Detection result on a congested traffic scene. (b) Detection result on far vehicles.

IV. Experiments

Our proposed approach is evaluated on the vehicle detection task of the KITTI object detection benchmark [7]. This benchmark originally aims to evaluate object detection of vehicles, pedestrians and cyclists from images. It contains not only image data but also corresponding Velodyne 64E range scan data. The groundtruth labels include both 2D object bounding boxes on images and its corresponding 3D bounding boxes, which provides sufficient information to train and test detection algorithm on range scans. The KITTI training dataset contains 7500+ frames of data. We randomly select 6000 frames in our experiments to train the network and use the rest 1500 frames for detailed offline validation and analysis. The KITTI online evaluation is also used to compare the proposed approach with previous related works.

For simplicity of the experiments, we focus our experiemts only on the *Car* category of the data. In the training phase, we first label all 3D points inside any of the groundtruth car 3D bounding boxes as foreground vehicle points. Points from objects of categories like *Truck* or *Van* are labeled to be ignored from $\mathcal{P}$ since they might confuse the training. The rest of the points are labeled as background. This forms the label $l_{\mathbf{p}}$ in (5). For each foreground point, its belonging bounding box is encoded by (4) to form the label $\mathbf{b}'_{\mathbf{p}}$ in (6).

The experiments are based on the Caffe [12] framework. In the KITTI object detection benchmark, images are captured from the front camera and range scans percept a 360° FoV of the environment. The benchmark groundtruth are only provided for vehicles inside the image. Thus in our experiment we only use the front part of a range scan which overlaps with the FoV of the front camera.

The KITTI benchmark divides object samples into three difficulty levels according to the size and the occlusion of the 2D bounding boxes in the image space. A detection is accepted if its image space 2D bounding box has at least 70% overlap with the groundtruth. Since the proposed approach naturally predicts the 3D bounding boxes of the vehicles, we evaluate the approach in both the image space and the world space in the offline validation. Compared to the image space, metric in the world space is more crucial in the scenario of autonomous driving. Because for example many navigation and planning algorithms take the bounding box in world space as input for obstacle avoidance. Section IV-A describes the evaluation in both image space and world space in our offline validation. In Section IV-B, we compare the proposed approach with several previous range scan detection algorithms via the KITTI online evaluation system.

TABLE I
Performance in Average Precision and Average Orientation Similarity for the Offline Evaluation

	Easy	Moderate	Hard
Image Space (AP)	74.1%	71.0%	70.0%
Image Space (AOS)	73.9%	70.9%	69.9%
World Space (AP)	77.3%	72.4%	69.4%
World Space (AOS)	77.2%	72.3%	69.4%

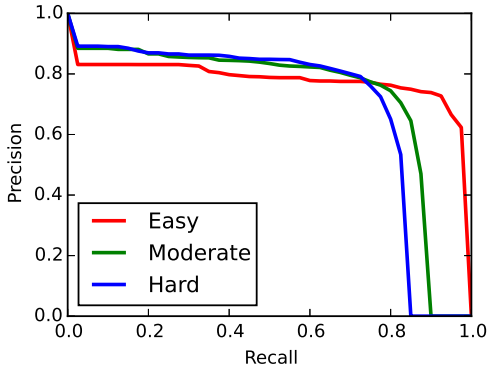


Fig. 5. Precision-recall curve in the offline evaluation, measured by the world space criterion. See Section IV-A.

A. Performane Analysis on Offline Evaluation

We analyze the detection performance on our custom offline evaluation data selected from the KITTI training dataset, whose groundtruth labels are accessable to public. To obtain an equivalent 2D bounding box for the original KITTI criterion in the image space, we projected the 3D bounding box into the image space and take the minimum 2D bounding rectangle as the 2D bounding box. For the world space evaluation, we project the detected and the groundtruth 3D bounding boxes onto the ground plane and compute their overlap. The world space criterion also requires at least 70% overlap to accept a detection. The performance of the approach is measured by the Average Precision (AP) and the Average Orientation Similarity (AOS) [7]. The AOS is designed to jointly measure the precision of detection and orientation estimation.

Table I lists the performance evaluation. Note that the world space criterion results in slightly better performance than the image space criterion. This is because the user labeled 2D bounding box trends to be tighter than the 2D projection of the 3D bounding boxes in the image space, especially for vehicles observed from their diagonal directions. This size difference diminishes the overlap between the detection and the groundtruth in the image space.

Like most detection approaches, there is a noticeable drop of performance from the easy evaluation to the moderate and hard evaluation. The minimal pixel height for easy samples is 40px. This approximately corresponds to vehicles within 28m. The minimal height for moderate and hard samples is 25px, corresponding to minimal distance of 47m. As shown in Figure 4 and Figure 1, some vehicles farther than 40m are scanned by very few points and are even difficult to recognize for human. This results in the performance drop for moderate and hard evalutaion.

Figure 5 shows the precision-recall curve of the world space criterion as an example. Precision-recall curves of the other criterion are similar and omitted here. Figure 4a shows the detection result on a congested traffic scene with more than 10 vehicles in front of the lidar. Figure 4b shows the detection result cars farther than 50m. Note that our algorithm predicts the completed bounding box even for vehicles which are only partly visible. This significantly differs from previous proposal-based methods and can contribute to stabler object tracking and path planning results. For the easy evaluation, the algorithm detects almost all vehicles, even occluded. This is also illustrated in Figure 5 where the maximum recall rate is higher than 95%. The approach produces false-positive detection in some occluded scenes, which is illustrated in Figure 4a for example.

B. Related Work Comparison on the Online Evaluation

There have been several previous works in range scan based detection evaluated on the KITTI platform. Readers might find that the performance of these works ranks much lower compared to the state-of-the-art vision-based approaches. We explain this by two reasons. First, the image data have much higher resolution which significantly enhance the detection performance for far and occluded objects. Second, the image space based criterion does not reflect the advantage of range scan methods in localizing objects in full 3D world space. Related explanation can also be found from Wang and Posner [31]. Thus in this experiments, we only compare the proposed approach with range scan methods of Wang and Posner [31], Behley et al. [2], Plotkin [22]. These three methods all use traditional features for classification. Wang and Posner [31] performs a sliding window based strategy to generate candidates and Behley et al. [2], Plotkin [22] segment the point cloud to generate detection candidates.

TABLE II
PERFORMANCE COMPARISON IN AVERAGE PRECISION AND AVERAGE ORIENTATION SIMILARITY FOR THE ONLINE EVALUATION

		Easy	Moderate	Hard
Image Space (AP)	Proposed	60.3%	47.5%	42.7%
	Vote3D	56.8%	48.0%	42.6%
	CSoR	34.8%	26.1%	22.7%
	mBoW	36.0%	23.8%	18.4%
Image Space (AOS)	Proposed	59.1%	45.9%	41.1%
	CSoR	34.0%	25.4%	22.0%

Table II shows the performance of the methods in AP and AOS reported on the KITTI online evaluation. The detection AP of our approach outperforms the other methods in the easy task, which well illustrates the advantage of CNN in representing rich features on near vehicles. In the moderate and hard detection tasks, our approach performs with similar AP as Wang and Posner [31]. Because vehicles in these tasks consist of too few points for CNN to embed complicated features. For the joint detection and orientation estimation evaluation, only our approach and CSoR support orientation estimation and our approach significantly wins the comparison in AOS.

V. CONCLUSIONS

Although attempts have been made in a few previous research to apply deep learning techniques on sensor data other than images, there is still a gap inbetween this state-of-the-art computer vision techniques and the robotic perception research. To the best of our knowledge, the proposed approach is the first to introduce the FCN detection techniques into the perception on range scan data, which results in a neat and end-to-end detection framework. In this paper we only evaluate the approach on 3D range scan from Velodyne 64E but the approach can also be applied on 3D range scan from similar devices. By accumulating more training data and design deeper network, the detection performance can be even further improved.

VI. ACKNOWLEDGEMENT

The author would like to acknowledge the help from Ji Liang, Lichao Huang, Degang Yang, Haoqi Fan and Yifeng Pan in the research of deep learning. Thanks also go to Ji Tao, Kai Ni and Yuanqing Lin for their support.

References

[1] Jens Behley, Volker Steinhage, and Armin B Cremers. Performance of Histogram Descriptors for the Classification of 3D Laser Range Data in Urban Environments. *2012 IEEE International Conference on Robotics and Automation*, pages 4391–4398, 2012.

[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] Ross Girshick, Jeff Donahue, Trevor Darrell, U C Berkeley, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Cvpr'14*, pages 2–9, 2014.

[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] 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.

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

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *ACM Multimedia*, 2:4, 2014.

[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] J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010.

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

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

[20] 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.

[21] 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.

[22] Leonard Plotkin. Pydriver: Entwicklung eines frameworks für räumliche detektion und klassifikation von objekten in fahrzeugumgebung. Bachelor's thesis (Studienarbeit), Karlsruhe Institute of Technology, Germany, March 2015.

[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 Pretrained Convolutional Neural Network Features. *IEEE International Conference on Robotics and Automation (ICRA)*, (May), 2015.

[25] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks. *arXiv preprint arXiv:1312.6229*, pages 1–15, 2013.

[26] 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.

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

[28] 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.

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

[30] 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.

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

[32] 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.

[33] 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.