EE 367 Report: Replacing LIDAR – 3D Object Detection using Monocular Depth Estimation

Benjamin Goeing
Department of Electrical Engineering
Stanford University
bgoeing@stanford.edu

Lars Jebe
Department of Electrical Engineering
Stanford University
larsjebe@stanford.edu

Abstract

This work investigates the potential and limitations of removing LIDAR sensors from 3D object detection systems. Such systems usually rely on both RGB images and corresponding LIDAR depth measurements as input. In our work, we replace the LIDAR depth input with a depth estimate produced by a neural network. Starting from a single RGB image, we obtain a depth map, convert it into a LIDAR-like point cloud and feed it into a second neural network for 3D object detection. This problem is fundamentally ill-posed because of the scale ambiguity that is inherent to monocular depth estimation. While our approach falls short of reaching near-LIDAR quality, the results are promising for cases in which the scale can be estimated accurately.

1. Introduction

LIDAR systems are currently irreplaceable for many autonomous driving and robotics applications. While the depth data is often good and reliable, LIDAR sensors tend to be very expensive, and there is a high demand for cheaper solutions. Our work attempts to produce reliable 3D Object detection without LIDAR system. In order to achieve this goal, we extend an existing 3D Object Detection System [11], by replacing the LIDAR depth input data with a monocular depth estimation from a neural network [5].

2. Related Work

3D Object Detection for autonomous driving applications is not a new problem and every year several new approaches are published. We are evaluating and comparing our results on the KITTI 3D Object Detection Dataset [4]. The approaches that are relevant to our method can be divided into two categories: Those that make use of active range scanning (LIDAR) and those that only use a single RGB image to perform the same task. Our approach falls into the latter category. Generally, the methods that use active range scanning vastly outperform the approaches in the second category. While [3], [14] and [11] obtain a 3D detection AP of $55\,\% - 85\,\%$, methods that don't use LIDAR, such as [1], [13], and [12] obtain a detection AP of $< 14\,\%$. Even methods using stereo information [2] are inferior to the ones using LIDAR.

Our method aims at combining a state-of-the-art approach for 3D object detection that makes use of LIDAR, with a monocular depth estimation network to replace the LIDAR input. We use [11] for the object detection and a weakly supervised approach for depth estimation from a single RGB image [5]. The choice of those networks was motivated by the performance of those models and the availability of their implementation. There are a lot of other well performing monocluar depth estimation methods [8], [9], [10], all of which suffer from the problem of scale-ambiguity. The scale ambiguity can be reduced to a minimum when training on a relatively homogeneous dataset (e.g. consisting of images with a fixed focal length), but it can never be eliminated completely. [6] partially overcomes this limitation by embedding knowledge of the focal length into the estimation process. A performance evaluation of recent monocular depth estimation can be found in [7].

3. New Technique

The existing Frustum PointNet model [11] takes an RGB image as well as a LIDAR point cloud as input modalities in order to produce 3D bounding boxes 1. We propose a new technique which gets rid of the LIDAR data entirely. Instead, we replace the depth input with a monocular depth-estimation from an RGB Image. Figure 2 visualizes this new approach. In a first step, we produce a depth map from an RGB image, using a convolutional neural network [5]. We then take this depth map, and transform it into a 3D point cloud to simulate depth information that was previously provided by the LIDAR. Lastly, we pass this point cloud into the Frustum PointNet to produce 3D bounding

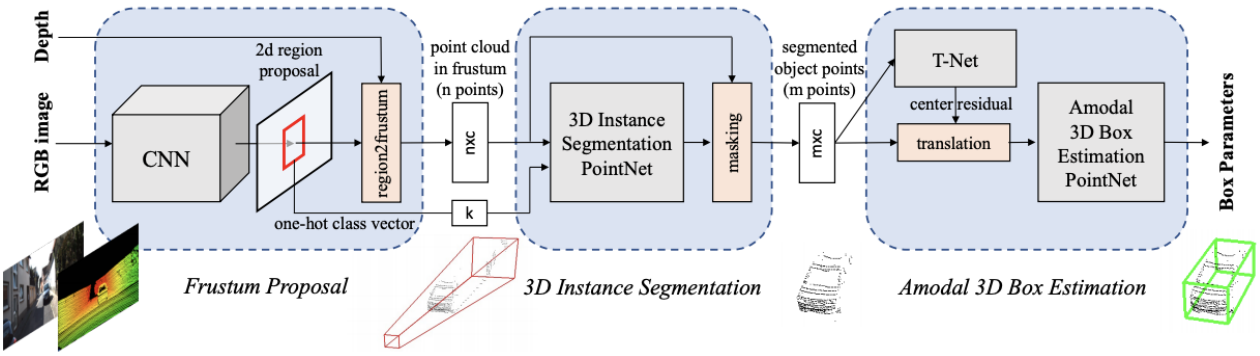

Figure 1. Existing Frustum Pointnet model architecture [11]

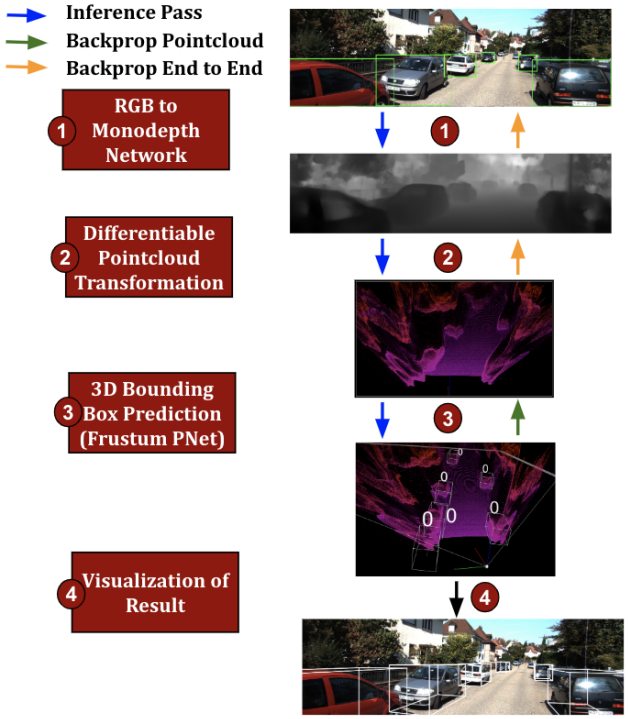

Figure 2. Our proposed architecture

boxes.

Due to the different structure and scale of the new depth information, the Frustum PointNet initially performed very poorly. We therefore launched an effort to re-train the model from the point cloud using the new input modality, and experimented with a variety of regularization and other hyper parameters. The results of this approach are detailed in the evaluation section. This approach is represented by the green arrow labeled "Backprop Point Cloud" in 2).

To take this approach one step further, we started the significant undertaking of making the entire dataflow trainable end-to-end, by integrating the two existing networks into a single trainable graph. A key challenge in accomplishing this was to rewrite the Point Cloud Transformation Operation (Step 2 in Figure 2), in order to make complete graph differentiable for backpropagation. In addition, the existing Frustum PointNet 2) heavily relied on offline preprocessing. We therefore had to additionally rewrite the entire data processing and dataloader classes, in order to enable this new end-to-end training. The data processing involved transform the disparity map to a real-world point cloud, enforcing constraints on the distance of the road to the camera (given by a fixed camera setup), and labelling points in the 3D frustum (defined by a 2D bounding box). This approach is represented by the orange arrows labelled "Backprop End to End" in 2). The results of this end-to-end approach are detailed in the evaluation section.

4. Evaluation and Results

Given the new input modality, the model initially performs poorly, with a 3D Box IoU below $5\,\%$. Through significant retraining using the point cloud input, our model achieves an IoU of $23.0\,\%$. This is far below the LIDAR-based IoU of $69.9\,\%$. However, given the absence of any real depth information, this approach shows some potential, and encourages further research. In particular, we noticed that the depth maps produced by the RGB-to-Monodepth network often look plausible, but lack the accuracy of an active range scanning system.

Figures 3 and 4 illustrate this. While the car is clearly recognizable in the depth map and in the point cloud, the depth map may suggests the car is 12 meters away, whereas it might actually only 10 meters away. This discrepancy alone is enough to produce an IoU of near zero. This fundamentally ill-posed problem of producing real depth information from a Monodepth stream is further detailed in the discussion section. A stereo-RGB-based depth estimator might produce much better results.

	FPointNet [11] (LIDAR)	**FPointNet + Monodepth** (Mono)	**End-to-end** (Mono)
Segmentation Accuracy (%)	90.4	71.5	80.8
3D Box IoU (%)	69.6	23.0	11.9
Detection AP (%) (IoU = 0.7)	62.9	3.5	0.9

Table 1. Comparison of the baseline method (left), our method only re-training the object detection network (middle) and training both modules end-to-end (right)

	Mono3D [1]	**MF3D** [13]	**MonoGRNet** [12]
3D Detection AP (%) (easy)	2.53	2.31	2.31
3D Detection AP (%) (medium)	10.53	5.69	5.39
3D Detection AP (%) (hard)	13.88	10.19	7.62

Table 2. Other single image 3D object detection methods. Our method obtains $3.5\,\%$ as an average across images in the three categories easy, medium, and hard.

Figure 3. Sample depth map produced

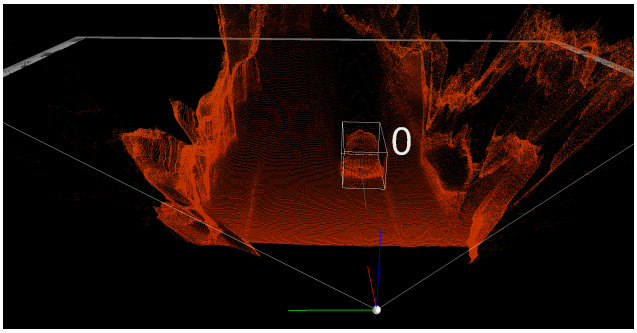

Figure 4. Sample 3D pointcloud produced

In a second run, we connected both the depth estimation and the 3D object detection network into one big tensorflow graph, and trained both in an end-to-end fashion. We surprisingly found that the end-to-end trained model does not perform on par with the fine-tuned model from above yet. This is most likely an implementation problem and discussed in the Future Work section. Table 1 shows the results from our efforts compared to the baseline model that uses LIDAR as input. The evaluation metric that is reported most often is the detection AP, or the percentage of predicted bounding boxes with an IoU above $70\,\%$. While our model achieves $3.5\,\%$ on this task, it is noteworthy that even state of the art methods perform poorly in this category, with detection APs that are usually below $10\,\%$. We have included the detection AP results of 3D object classification methods that use a single RGB image as input in table 2.

The fact that the end-to-end model achieves a higher segmentation accuracy but a lower 3D Box IoU and detection AP is due to the depth maps that are produced in the process of running the model. As further detailed below, we observe the depth maps to be inferior in the end-to-end case due to some implementation problem. This leads to fewer points in the point cloud being labelled as part of the object, because most points of the predicted object will lie outside the 3D bounding box. The percentage of points that are not part of the object is thus higher, and it's easier to predict that a point is not part of the object. However, accurately predicting the 3D bounding box becomes much more challenging given incorrect depth information, which is why we observe a lower box IoU and detection AP.

5. Discussion

While our approach fall short of reaching near-LIDAR performance, it merits further research in this direction. With more time for training and for fine-tuning hyper parameters we expect further performance improvements. Given the problems that arise form monocular depth estimation, we conclude that a similar approach, enhanced with some additional prior depth information, e.g. from stereo images or a cheap, sparse LIDAR system, would be helpful.

5.1. Limitations

The results show that a single RGB is not enough to reach near-LIDAR performance for the task of 3D object detection. The largest factor that contributes to the loss in performance is the scale ambiguity that is inherent to monocular depth estimation. We observe that the predicted

depth maps are often plausible, yet the scale and distance of the objects does not correspond to the ground truth. For example, we frequently observe that a car at a distance of 20 meters is estimated to be 18m or 22m away. This relatively small error of $10\,\%$ leads to a significant drop in the performance, which is measured as intersection over union of the 3D bounding boxes of cars, cyclist and pedestrians in the KITTI dataset. Given a more reliable method for depth estimation, such as calibrated stereo correspondences or even a very spare point cloud from a relatively cheap active range scanning system could potentially increase the performance of 3D object detection significantly while keeping the overall cost of the system relatively low.

6. Future Work

Our model is training end-to-end, however, the results from the end-to-end trained model are not convincing yet. The end-to-end trained model should at least perform on a similar level as the model that used the monocular depth estimation in inference only to fine-tune the Frustum PointNet. In order to make the depthmap to pointcloud transformation differentiable for end-to-end training, we had to reimplement a number of methods from scratch in tensorflow. We have benchmarked invdividual components of this transformation against the non-differentiable equations, and have found that the resize-methods of Tensorflow and OpenCV are producing very different results to the extent, that the depth estimation can be off by up to a few meters. This is a known issue, which the tensorflow team is currently trying to resolve (last post was 20 days ago) `https://github.com/tensorflow/tensorflow/issues/6720`)/ This difficulty significantly diminishes the quality of a pre-trained network, and additionally also invalidates depth prior estimates, which worsens the ability of the depth estimation to predict depth at the correct scale. A retraining on the monodepth network using input images that were resized with Tensorflow instead of OpenCV would improve this.

In addition, currently only the final loss of the object detection is backpropagated through the network. To improve training, it might be helpful to add intermediate stages (such as the result of the depth estimation) to the total loss. As discussed in 5.1, a more viable approach to 3D object detection is using a depth estimation method that can predict depth at the correct scale (such as depth from calibrated stereo-images). This approach is worth trying, since it overcomes one fundamental challenge while retaining the main value-proposition of a low-cost system.

References

[1] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.

[2] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[3] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[6] L. He, G. Wang, and Z. Hu. Learning depth from single images with deep neural network embedding focal length. *IEEE Transactions on Image Processing*, 27(9):4676–4689, 2018.

[7] T. Koch, L. Liebel, F. Fraundorfer, and M. Körner. Evaluation of cnn-based single-image depth estimation methods. In *European Conference on Computer Vision*, pages 331–348. Springer, 2018.

[8] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016.

[9] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3372–3380, 2017.

[10] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[11] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[12] Z. Qin, J. Wang, and Y. Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. *arXiv preprint arXiv:1811.10247*, 2018.

[13] F. Tung and J. J. Little. Mf3d: Model-free 3d semantic scene parsing. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4596–4603. IEEE, 2017.

[14] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.