Methods

András Rövid, Viktor Remeli* and Zsolt Szalay

Raw fusion of camera and sparse LiDAR for detecting distant objects

Fusion von Kameradaten und spärlichem LiDAR-Rohsignal zur Erkennung entfernter Objekte

https://doi.org/10.1515/auto-2019-0086
Received July 25, 2019; accepted March 17, 2020

Abstract: Environment perception plays a significant role in autonomous driving since all traffic participants in the vehicle's surroundings must be reliably recognized and localized in order to take any subsequent action. The main goal of this paper is to present a neural network approach for fusing camera images and LiDAR point clouds in order to detect traffic participants in the vehicle's surroundings more reliably. Our approach primarily addresses the problem of sparse LiDAR data (point clouds of distant objects), where due to sparsity the point cloud based detection might become ambiguous. In the proposed model each 3D point in the LiDAR point cloud is augmented by semantically strong image features allowing us to inject additional information for the network to learn from. Experimental results show that our method increases the number of correctly detected 3D bounding boxes in sparse point clouds by at least 13–21 % and thus raw sensor fusion is validated as a viable approach for enhancing autonomous driving safety in difficult sensory conditions.

Keywords: neural networks, sensor fusion, autonomous driving, LiDAR, sparse point cloud

Zusammenfassung: Die Wahrnehmung der Umgebung spielt beim autonomen Fahren eine wichtige Rolle, da alle Verkehrsteilnehmer in der Umgebung des Fahrzeugs zuverlässig erkannt und lokalisiert werden müssen, um weitere Maßnahmen ergreifen zu können. Das Hauptziel dieses Artikels ist es, eine neuronale Netzwerk Methode zur Integration von Kamerabildern und LiDAR-Punktwolken vorzustellen, um Verkehrsteilnehmer in der Fahrzeugumgebung zuverlässiger zu erkennen. Unser Ansatz befasst sich hauptsächlich mit dem Problem spärlicher LiDAR-Daten (Punktwolken entfernter Objekte), bei denen die auf Punktwolken basierende Erkennung aufgrund der Sparsamkeit möglicherweise mehrdeutig wird. Im vorgeschlagenen Modell wird jeder 3D-Punkt in der LiDAR-Punktwolke durch semantisch starke Bildmerkmale erweitert, sodass wir zusätzliche Informationen anfügen können, aus denen das Netzwerk lernen kann. Experimentelle Ergebnisse zeigen, dass unsere Methode die Anzahl korrekt erkannter 3D-Begrenzungsrahmen in spärlichen Punktwolken um mindestens 13–21 % erhöht. Daher wird die rohe Sensorfusion als praktikabler Ansatz zur Verbesserung der autonomen Fahrsicherheit unter schwierigen sensorischen Bedingungen validiert.

Schlagwörter: Neuronale Netze, Sensorfusion, autonomes Fahren, LiDAR, spärliche Punktwolke

1 Introduction

Sensing and understanding vehicle surroundings is one of the most crucial factors in autonomous driving, since any subsequent action taken is strongly dependent on how the scene is interpreted, what type of participants are present, where they are located, what their intention is, etc. In order to make self-driving safe and reliable all this information must be extracted and more importantly must be accurate. Any misdetected or misclassified object may harm the self-driving safety. One way to increase the reliability of perception is the utilization of various types of sensors. In heterogeneous sensor setups the overall joint sensing capability of the self-driving vehicle covers a wider range of weather and traffic conditions in general, furthermore the sensor redundancy is also a significant advantage in case of sensor failure, damage, or obstruction.

The benefits of redundant and especially multi-modal sensor setups come at the cost of a different challenge, namely how multi-sensor data might be fused in order to detect and recognize traffic participants reliably. The two

***Corresponding author: Viktor Remeli,** Department of Automotive Technologies at the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary, e-mail: viktor.remeli@gjt.bme.hu
András Rövid, Zsolt Szalay, Department of Automotive Technologies at the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary, e-mails: andras.rovid@gjt.bme.hu, zsolt.szalay@gjt.bme.hu

main approaches to multi-sensor integration are object-level and low-level (raw) data fusion. While in case of object level fusion the individual sensor signals are independently processed, yielding usually an object list for each individual sensor as described by Hall and Llinas [3] and Kim et al. [5], in case of low-level fusion the detection task benefits from low-level features of all sensors jointly. In contrast to various high-level fusion approaches (e. g., track-to-track fusion in Kovacs et al. [6]), low-level fusion and raw fusion have received somewhat less attention. We believe it is likely that with purely high-level fusion we are not leveraging the full range of inter-sensor synergies present in most real-world setups, and are therefore not achieving a perception performance otherwise possible.

In this work we explore the possible advantages of lower level data fusion. Due to different low-level representations of data and their different levels of spatial and temporal sparsity (coming from different types of sensors), joint data representation methods are essential. An important consideration here is how to fuse data without substantial information loss or artificial data imputation.

Let us consider the case of integrating a camera image and a sparse LiDAR point cloud in an RGBD (red-green-blue-depth) matrix. By reprojecting all point cloud points onto the camera image there might be many pixels with one or more reprojected 3D points but at the same time there would certainly also be many pixels with no reprojected 3D points at all. Also, certain 3D points relevant to the detection might be reprojected just outside the boundaries of the image raster. How can the two types of data be jointly represented in this particular case without information loss or artificial injection? Simple oversampling and interpolation techniques are clearly not adequate in this regard, as they would necessarily introduce an artificial and highly inaccurate depth model into our calculations.

Instead of fusing camera and LiDAR data in the image plane, the bird's eye view (BEV) plane, or in an arbitrarily voxelized 3D space, we choose to avoid introducing artificial structures and assumptions. Therefore we train our networks on an augmented version of the original, raw point cloud representation that lacks spatial structure. Our contribution lies in the novel method of point cloud augmentation that accomplishes a meaningful data fusion between camera and LiDAR features and allows a much closer degree of sensor integration than previous point cloud based approaches.

Our results show that it is possible to achieve more accurate and more reliable perception by utilizing appropriate methods of data fusion and thus leveraging the synergies hidden in the statistical association between streams of data provided by multiple sensors that simultaneously measure the same environment. In particular we achieve a tangible 13–21 % increase in the number of correct detections in sparse point clouds. The practical significance of our result lies in the possibility of more reliably detecting distant traffic participants or obstacles and thus achieving a higher level of safety for autonomous driving.

2 Related work

In the following section we present a brief overview of some relevant state of the art results achieved in 3D object detection for autonomous driving (trained, validated and tested on the KITTI dataset[1]).

Ku et al. [7] propose an aggregate view object detection network (AVOD) for autonomous driving scenarios. The authors use LiDAR point clouds and RGB images to generate features that are shared by two subnetworks. In case of LiDAR data a six channel bird's eye view (BEV) map is considered. Liang et al. [9] also exploit both LiDAR and cameras. With the help of a so called "continuous fusion layer" a dense BEV feature map is created and fused with the BEV feature map extracted from LiDAR. Although the BEV map is a convenient way to overcome the representation related differences of camera image and LiDAR point clouds, the LiDAR data is not fully utilized.

In this work we employ the PointNet neural network architecture which was introduced by Qi et al. [13] and was purpose-made for processing unstructured collections of data points. Previous work has already shown this architecture to be a capable alternative for 3D detection of traffic participants, but to the best of our knowledge we are the first to demonstrate the low-level sensor fusion and its benefits that become available in this setup.

A straightforward, early application of PointNet is F-PointNet which is basically a two-stage 3D detector proposed by Qi et al. [12] and evaluated on the KITTI dataset. In the first stage the RGB camera image is processed by an arbitrary object detector that determines the 2D detection boxes and the corresponding object types. In the second stage each 2D box is used to form a 3D bounding frustum in order to reduce the point cloud to the 3D points of interest only. The set of points falling into the frustum is then further processed by two PointNet networks in order to establish the 3D bounding box coordinates (position, size and heading). It is worth noting that although this setup relies

1 For further information, see Geiger, Lenz and Urtasun [2] and Geiger et al. [1].

on both camera and LiDAR data, the fusion of the inputs is rather high-level and loose, as the 3D LiDAR points are combined with only the 2D box coordinates and the object type prediction, but not the detailed pixel-level information available from the RGB image. Another PointNet-based design is given by Wang and Jia [16] who also use a 2D detector in the first stage, but in the second stage they work with a sequence of frustums instead of a single frustum, each one yielding a single embedding vector via a PointNet. Appropriate concatenation of these vectors forms a feature map from which the final prediction is derived by a 2D convolution followed by a classification and regression head. Again, the camera image contributes only the 2D bounding box, but no low-level features. Our work differs from both above mentioned frustum-based approaches in that we recognize and pursue the possibility of deepening the level of fusion in order to fully exploit the benefits of inter-sensor synergies.

Xu, Anguelov and Jain's [17] model is also based on the PointNet architecture and thus able to handle unorganized raw point cloud data. They combine point cloud features learned by a PointNet with high-level features – extracted by a ResNet[2] based CNN – that globally describe the 2D window containing the object. The authors utilized the image feature vector taken from the semantically highest level of ResNet (1×1×2048 in shape) and have extended each point cloud data point with the same global descriptor. The data augmentation they employ thus describes the object as a whole: their approach differs from ours in that they do not extract the local characteristics of the object. Unlike Xu et al., we assign local and specific image features to the corresponding LiDAR points.

PointNet has proven to be a popular and quite successful design, with several PointNet-based architectures repeatedly achieving top-10 status on the KITTI 3D object detection benchmark.[3] Besides the three already described approaches, we would like to mention three further PointNet-based architectures that achieve good results, albeit working with a single sensor (LiDAR) only. Zhou and Tuzel [18] break up the point cloud into voxels, each non-empty voxel is then represented by a fixed-length vector via a PointNet. A subsequent sparse 3D convolution and a region proposal network convert this intermediate representation into the final 3D bounding box predictions. Shi, Wang and Li [15] perform a foreground/background segmentation first, and then create 3D box proposals for each of the foreground points, which are then later refined into the final predictions. Lang et al. [8] learn a pillar-wise feature representation for the point cloud, which then only has to be processed by a 2D convolution, resulting in significant inference speed-up without losing precision. It is possible that these and similar LiDAR-only PointNet architectures could benefit the most from integrating camera information the way we propose in this paper. Of course, practical feasibility with regards to run time would have to be investigated as the inclusion of a second information source is bound to increase overall computational complexity. Since we demonstrate tangible advantages of our approach on distant objects only, it could even make sense to only partially augment the point cloud by exclusively focusing on distant regions of interest – which would usually contain far less points than the closer regions. Although not an immediate concern in this paper, these kinds of questions could be explored in further studies.

3 Proposed architecture

3.1 Problem definition

Our goal is to enhance environment perception and autonomous driving safety by utilizing raw image data and LiDAR point clouds jointly without significant information loss, favoring the direct processing of unorganized LiDAR point clouds instead of relying on birds eye view images or other projections or voxelizations that cause data degradation. The PointNet architecture described in Section 2 seems to be a promising and popular design for handling unorganized point cloud data by deep neural network architectures directly – the question we try to answer is whether a low-level fusion is viable in such a setup.

The primary use-case we focus on is 3D object detection in sparse LiDAR point clouds corresponding to a multi-sensor autonomous driving scenario with distant traffic participants.

In order to be able to argue about *sparse* point clouds and *distant* objects, we have to define what we mean by those words. In general, we will consider point clouds that fall into an object's frustum and that consist of 8 or fewer points *sparse*. The vehicle-object distance that corresponds to a sparse point cloud is dependent on both the object and the LiDAR type. We will consider two main object types and three LiDAR types for this discussion. The two ideal object types considered are the *single-row* (squat, i. e., vehicle-like) and the *two-row* (tall, i. e., pedestrian-like) object configuration. A vehicle-like object's bound-

2 For details on ResNet architecture, consult He et al. [4].

3 For benchmark results, see: http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d

Table 1: Object distances.

Sensor	64-channel LiDAR	32-channel LiDAR	16-channel LiDAR
Vertical resolution	0.43°	1.29°	2.00°
Horizontal resolution	0.08°	0.24°	0.37°
Distant vehicle	213 m	71 m	46 m
Distant pedestrian	227 m	76 m	49 m

ing box is assumed to be 1.6 m high and 2.56 m wide and the pedestrian-like object's bounding box is assumed to be 1.7 m high and 0.68 m wide. If we consider a 2D grid made up of several (16, 32 or 64) horizontal LiDAR scans projected onto the camera plane, we will assume the vehicle-like object's bounding box spans 1×8 cells and is expected to contain 8 points in a single horizontal row while the pedestrian-like object will span 2×4 cells containing 8 points in two rows. Note that the horizontal resolution of the LiDAR grid is much higher than the vertical: in this case we will assume the azimuth (horizontal resolution angle) of the devices is always $\frac{1}{5.375}$-th of the vertical resolution angle. According to our working hypotheses it is easy to calculate that *distant* objects resulting in *sparse* point clouds will typically occur starting at distances between 46 and 213 m, depending on LiDAR device, as can be seen in Table 1.

A depiction of LiDAR points on a vehicle-like object is given in Figure 1. It can be seen that determining the 3D bounding box based only on the points' 3D coordinates is not a trivial task even if the type of object is known in advance. The intuition behind our method is that the 3D bounding box estimation might become easier if we receive some additional information about the LiDAR points. Besides coordinates, it would also seem helpful to know certain additional semantic properties of a detected point: e. g., does it belong to a front bumper, a windshield, a tire, a leg, a head, etc. We suspect that the camera image, and in particular certain mid-level, localized image features from a 2D detector might contain the required information.

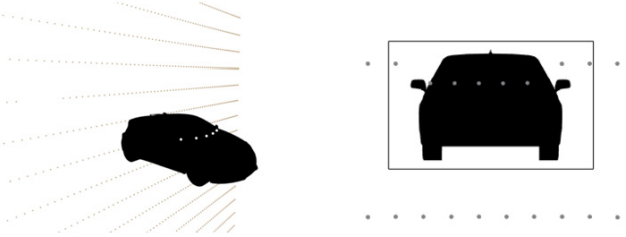


Figure 1: A 1.6×2.56 m vehicle-like bounding box with eight LiDAR points that fall into the frustum and a silhouette of a car in the background for reference. The image depicts a *distant* vehicle at 71 m detected using a 32-channel LiDAR.

3.2 Main contribution

Compared with other PointNet-based approaches, our work is unique in that it considerably deepens the level of integration between the two primary sensors, while also demonstrating the thus attainable performance benefits – especially in the case of sparse point clouds. The approach proposed in this paper extends and modifies the F-PointNet model by assigning local – yet semantically strong – image features from high resolution feature maps to each point in the LiDAR point cloud (for details refer to Section 3.3).

Our main contributions in this paper might be summarized as follows:

1. We augment each 3D raw LiDAR point with local features acquired from high resolution semantically strong image feature maps. Here we utilize the advantages offered by feature pyramid networks (FPNs) in order to acquire semantically strong feature maps at different scales.
2. We connect the augmented input to both the segmentation as well as the bounding box estimation part of the F-PointNet architecture. According to our experiments, this setup contributes to an improved accuracy.
3. We modify the internal structure of F-PointNet by increasing the filter depths for both segmentation and bounding box estimation subnetworks in order to expand the models' capacities in line with the increase of the input dimensions.
4. We show that this kind of augmentation benefits 3D detection in sparse point clouds. It appears that semantically strong local information about the 3D point contributes to an increased certainty in segmentation and 3D bounding box estimation. We have performed various experiments with sparse point clouds taken from the KITTI dataset in order to show that augmentation helps to increase the 3D detection accuracy.

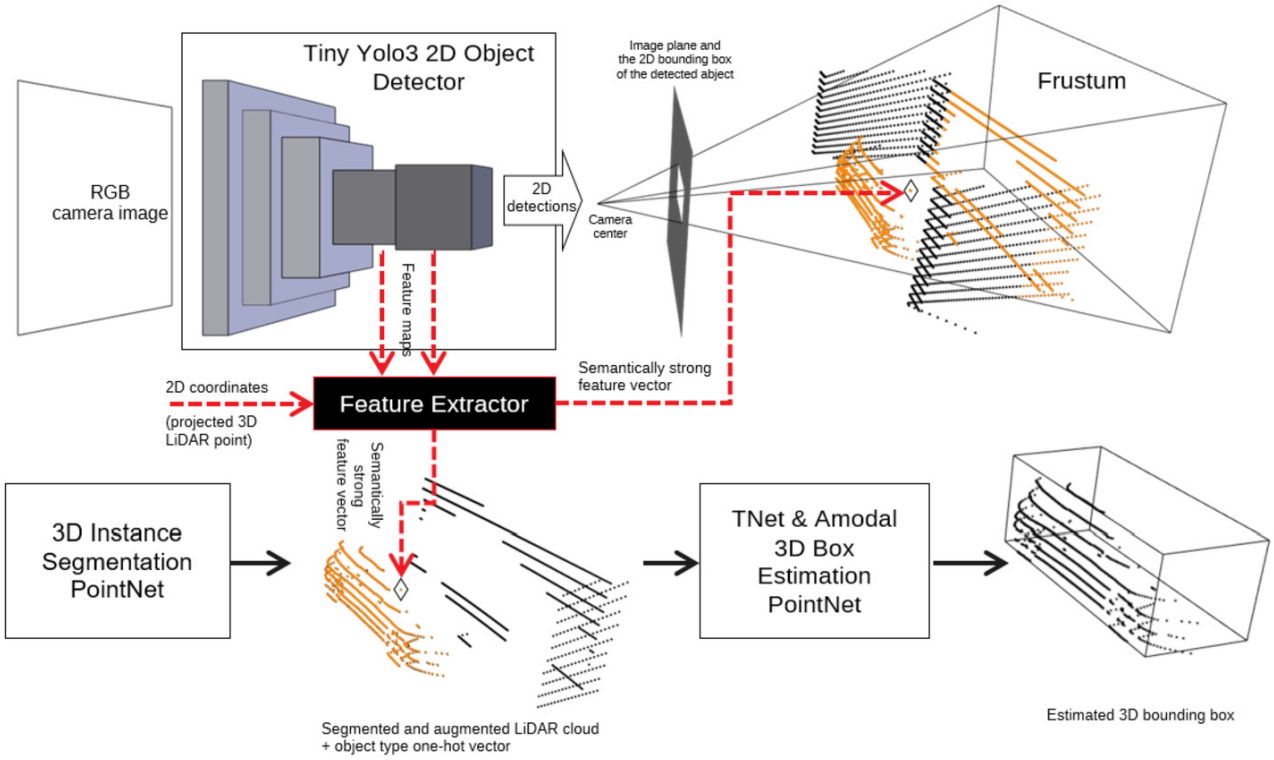


Figure 2: The proposed LIDAR-Camera fusion architecture. The point cloud augmentation block consists of a feature extractor that functions as indicated by the dashed lines.

3.3 Developed model

The overall model consists of three main parts, namely

- our custom-trained 2D detector network;
- the proposed point cloud augmentation block (see Fig. 2) which constitutes the main contribution of this paper;
- and finally the segmentation block and the bounding box estimation block: both parts correspond to the original F-PointNet design but had to be appropriately modified to accommodate the new setup.

The first stage of data processing is to identify the traffic participant objects and determine their 2D bounding boxes in the camera image, based on which the bounding 3D frustum will be established for each object. We use a 2D object detector which we have specifically trained for traffic participant detection. In particular, we have repurposed a Tiny Yolo3 [14] model that was pre-trained on the COCO dataset [11]. We conducted transfer learning for 20 epochs on the KITTI dataset aiming for more accurate Car, Pedestrian and Cyclist detection only. The training was done with the first half of the network's layers frozen throughout and the second half gradually opened up for training. We also make use of the advantages given by a feature pyramid design of the convolutions (FPN, Lin et al. [10]) in which high resolution semantically strong feature map representations are produced.

Prior to being processed by the segmentation and box-estimation PointNets, we expand each LiDAR point's 3D vector with semantically strong local image features taken from the 2D feature maps produced by the 2D detector. Given the calibration data of cameras as well as LiDAR the 3D LiDAR points are projected onto the camera image plane and the corresponding image feature vector is assigned to the 3D LiDAR point.

The reprojection of a 3D point to image coordinates is performed as follows. Assume that the camera matrices $\mathbf{P}_j = \mathbf{K}^{(j)}[\mathbf{R}_C^{(j)}|\mathbf{t}_C^{(j)}]$ and the extrinsics $[\mathbf{R}_L, \mathbf{t}_L]$ of the LiDAR wrt. the world frame are given. Here $\mathbf{R}_C^{(j)}$ and $\mathbf{t}_C^{(j)}$ denote the rotation and translation of the jth camera respectively (wrt. the world frame), $\mathbf{K}^{(j)}$ contains the intrinsics of the jth camera. Each 3D LiDAR point is projected onto the camera image plane of the second camera. Let the LiDAR point $\mathbf{X}_i$ be represented by a homogeneous 4-vector $[X_i, Y_i, Z_i, 1]$ in the world coordinate system, $i = 1..N$, where N stands for the number of points in the point cloud falling inside a given frustum defined by the camera center and the 2D detection window. Furthermore, let the image

points $\mathbf{x}_{ij}$ stand for the projections of $\mathbf{X}_i$ in the jth camera image given by a homogeneous 3-vector. By assuming the pinhole camera model, $\mathbf{x}_{ij}$ is expressed as follows: $\mathbf{x}_{ij} = \mathbf{K}^{(j)}[\mathbf{R}_C^{(j)}|\mathbf{t}_C^{(j)}]\mathbf{X}_i$. Because the camera images have been undistorted in advance in the pre-processing phase, radial and tangential lens distortions are not considered here.

Next, for all $\mathbf{X}_i$ the first M image features are taken from the 4th convolutional layer of the 2D detector. Although the 4th convolutional layer is not the one with the semantically strongest image features, it has a high resolution. Opting for a different tradeoff between semantics and resolution, features from other layers could have been chosen instead. According to our results however, the 4th layer contains features of appropriate semantic complexity to demonstrate the advantage of our proposed fusion method. The spatial resolution of the 4th layer is $8\times$ higher than the resolution of the semantically strongest feature map. In particular, each pixel from the 4th layer map corresponds to a 16×16 px patch on the original image. In order to assign features from the map to the individual reprojected LiDAR points, the map is first scaled up to the size of the original image.

Let us denote the tensor containing the semantically strong image feature maps from our 2D detector by tensor $\mathbf{F}$ of size $W \times H \times K$, where H and W correspond to map height and width, respectively and K stands for the number of feature maps in the given layer. Let $\mathbf{F}_k$ denote the feature map being at the kth position in $\mathbf{F}$, where $k \leq K$.

The task of the proposed augmentation block is to take local image feature vectors corresponding to the image projections of 3D LiDAR points (located inside the frustum defined by the camera intrinsics and 2D bounding box of the target) and assign them to the corresponding 3D point in the LiDAR point cloud by concatenation. Let us denote an augmented 3D point as:

$$\mathbf{X}_i^{aug} = [X_i, Y_i, Z_i, \mathbf{F}_1[\mathbf{x}_{ij}], \mathbf{F}_2[\mathbf{x}_{ij}], ..., \mathbf{F}_M[\mathbf{x}_{ij}]],$$

where $0 \leq M \leq K$, $\mathbf{F}_k[\mathbf{x}_{ij}]$ stands for a particular feature located at $\lfloor \mathbf{x}_{ij} \rfloor$ in the kth feature map.[4]

The augmented 3D point list is connected to the input of PointNet as well as to the input of the TNet networks (see Fig. 2) in the following form:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & \mathbf{F}_1[\mathbf{x}_{1j}] & ... & \mathbf{F}_M[\mathbf{x}_{1j}] \\ X_2 & Y_2 & Z_2 & \mathbf{F}_1[\mathbf{x}_{2j}] & ... & \mathbf{F}_M[\mathbf{x}_{2j}] \\ & & & & & \\ X_N & Y_N & Z_N & \mathbf{F}_1[\mathbf{x}_{Nj}] & ... & \mathbf{F}_M[\mathbf{x}_{Nj}] \end{bmatrix}$$

4 Experiment setup and results

We performed a comparative study on the impact of the proposed feature fusion upon precision scores versus the baseline score of the same (F-PointNet based) architecture without low-level fusion. The performance indicator used for evaluation was average precision (AP) calculated as defined in the official KITTI benchmark evaluation package, with a single difference. We performed our experiments using RGB image and *sparse* LiDAR point cloud inputs as defined in Section 3.1. In order to be able to perform many measurements, we artificially made all point clouds equally *sparse* by setting the sampling size parameter to 8 points per frustum. Note that the *sparse* sampling is a deliberate deviation from the standard KITTI evaluation method that makes the detection task much more challenging: the baseline AP decreases by an order of magnitude, signifying the difficulty of performing accurate 3D bounding box estimation in *sparse* (distant) point clouds.

Due to the stochastic nature of neural network training we opted for *Welch's unequal variances t-test* analysis of the difference of the scores of the two approaches. We trained 60 models altogether, 30 with the baseline setup (group A) and 30 with our proposed low-level fusion (group B). In our approach, we augmented the point clouds' 3 original channels $[X, Y, Z]$ with 29 features taken from medium-resolution but semantically relevant feature maps of our Tiny Yolo v3 2D detector. The 60 models were trained independently of each other, and given the randomization in weight initialization,[5] batch ordering, data augmentation and dropout, their performance metrics can also be regarded as independent random variables. Considering the large sample size the distribution of sample means (and their differences) can be treated as Gaussian due to the Central Limit Theorem. Therefore, *Welch's t-test* is applicable.

Our null hypothesis states that there is no difference upon applying our modification ($H_0 : \mu_B - \mu_A = 0$). The alternative hypothesis states that our modification indeed raises the AP performance for a given task ($H_1 : \mu_B - \mu_A > 0$). We will set the significance level at 5 %, and the corresponding confidence interval will be constructed with a confidence level of 90 %.

First we measured end-to-end performance of the systems and detected a statistically and practically significant AP improvement of 0.93 ± 0.69 percentage points (corresponding to 21 % more correct detections) in the *easy*

4 Instead of getting the feature from integer valued pixel coordinates bilinear interpolation could have also been used.

5 Our setup uses the tensorflow default Xavier uniform weight initializer.

Table 2: 90 % confidence intervals for achieved percentage point improvement in AP of camera and sparse LiDAR based 3D bounding box detection.

Using *Tiny Yolo v3 2D boxes*		3D car detection AP		
		easy	moderate	hard
Baseline	mean AP	4.40 %	4.22 %	3.89 %
	AP std	1.56 %	1.42 %	1.42 %
Our low-level fusion	mean AP	5.33 %	4.78 %	4.50 %
	AP std	1.65 %	1.56 %	1.56 %
Improvement	90 % CI center	*+0.93*	+0.57	+0.61
	90 % CI delta	*±0.69*	±0.64	±0.64
	$P(T_{DOF} \geq t)$	*1.41 %*	7.38 %	5.89 %
	Statistically significant	*yes*	no	no
	Relative improvement	*21.2 %*	13.4 %	15.8 %

Table 3: 90 % confidence intervals for achieved percentage point improvement in AP of camera and sparse LiDAR based 3D bounding box detection when using an ideal (ground truth) 2D detector. The effects of our low-level fusion method are more pronounced.

Using *ground truth 2D boxes*		3D car detection AP		
		easy	moderate	hard
Baseline	mean AP	6.47 %	7.34 %	6.97 %
	AP std	2.81 %	2.49 %	2.61 %
Our low-level fusion	mean AP	9.17 %	8.93 %	8.44 %
	AP std	2.20 %	2.54 %	2.59 %
Improvement	90 % CI center	*+2.70*	*+1.59*	*+1.47*
	90 % CI delta	*±1.09*	*±1.09*	*±1.12*
	$P(T_{DOF} \geq t)$	*0.01 %*	*0.87 %*	*1.65 %*
	Statistically significant	*yes*	*yes*	*yes*
	Relative improvement	*41.8 %*	*21.7 %*	*21.0 %*

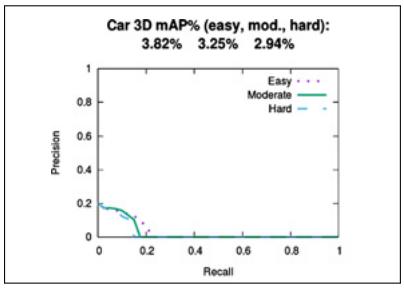
detection category[6] when using low-level fusion. We also found indication for the existence of a somewhat lesser improvement of around 0.6 percentage points in *moderate* and *hard* cases at a 10 % significance level. For details refer to Table 2 and Figures 3 and 4. Since the power of our test was calibrated to detect a 1 percentage point effect in 80 % of the cases and a 0.5 point effect in 34 % of the cases, we had good reason to suspect that the improvements in the *medium* and *hard* categories were indeed caused by our modification and not by chance, and that we would be committing a Type II error should we retain the null hypothesis in these cases.

In order to confirm our suspicion (without training and evaluating hundreds of networks) we repeated the experiments using actual ground truth – instead of predicted – 2D boxes in order to better isolate the effect of our modification. Thus we excluded the performance variation due to the 2D detection task of the first stage, which can be regarded as a fairly independent problem. Our measurements confirmed a very clear and statistically significant improvement in all three difficulty classes: our method achieved 1.47–2.70 percentage points higher AP scores corresponding to a performance increase of 21–42 % in the simplified task of correctly predicting a car's 3D bounding box given its sparse point frustum (not the whole point cloud) and a corresponding RGB image. Detailed results are available in Table 3 and Figures 3 and 4.

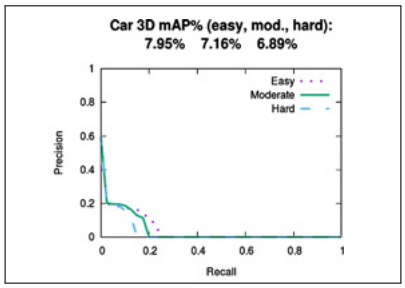
5 Summary

Our experiments indicate that low-level camera and sparse LiDAR data fusion is a viable option for improving perception in self-driving applications, where safety considerations play a central role. In this paper we propose a possible improvement for existing state of the art neural network architectures that consider camera and LiDAR

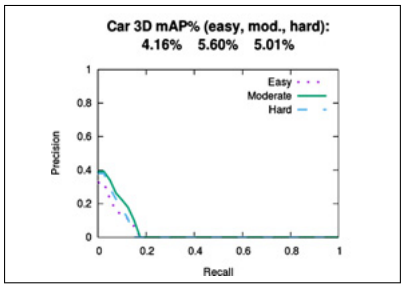
6 The KITTI 3D Object Detection Benchmark evaluates three separate AP scores for detecting objects that belong to one of three difficulty classes (*easy, moderate, hard*) as described in http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d



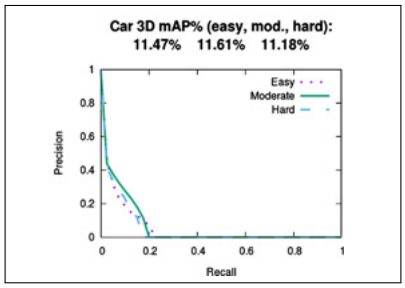
(a) Full detection task with baseline setup



(b) Full detection task with fusion setup

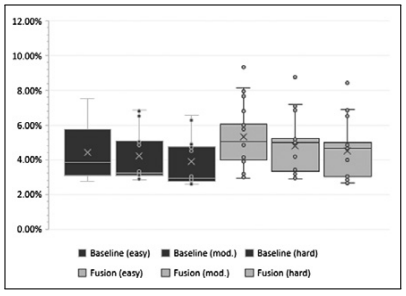


(c) Simplified detection task with baseline setup

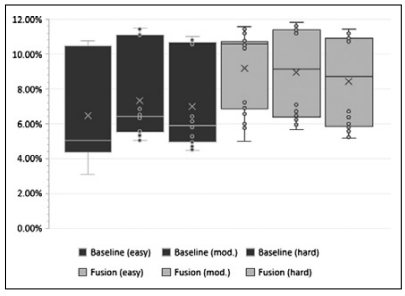


(d) Simplified detection task with fusion setup

Figure 3: Results of the model evaluation for sparse 3D point clouds consisting of 8 points. The figures compare the average precision (AP or equivalently mAP) scores of different models performing 3D bounding box detection for cars: (a) and (b) compare the performance of the baseline setup (where the cloud is represented as $[X, Y, Z]$ point records) with the proposed fusion setup (that augments the original $[X, Y, Z]$ point coordinates with 29 semantically strong image features) for the whole end-to-end detection task including both the 2D detection and the 3D detection phases. Subfigures (c) and (d) make the same comparison focusing only on the 3D detection subtask (assuming an ideally pre-solved 2D detection phase). (a) and (c) were measured using the same baseline model while (b) and (d) were produced using the same fusion model.



(a) Full detection task baseline vs. fusion



(b) Simplified detection task baseline vs. fusion

Figure 4: Average precision (AP) scores for 30 baseline networks and 30 fusion-enabled networks measured for the whole end-to-end detection task including both the 2D detection and the 3D detection phases (a); and for the simplified 3D detection subtask, assuming an ideally pre-solved 2D detection phase (b).

data only in separate, sequential or parallel phases of processing. The overarching concept behind our method is to use unstructured point cloud inputs that receive a semantically meaningful augmentation from an RGB detector. This input is then used both in training and deployment of various deep learning algorithms – e. g., 3D object detectors – that process raw signals from different types of sensors jointly (and not separately). We achieve this by projecting the LiDAR point cloud onto the image plane and augmenting the points with a corresponding image feature vector taken from the internal layers of the 2D detector.

Our results show that the proposed low-level fusion method can increase AP scores by at least 13–21 % in a *sparse* setting. We do not claim however that our fu-

sion method notably improves detection performance in a regular setting, nor that it is universally applicable in all pointcloud-based neural architectures. A performance comparison with the unmodified F-PointNet is sufficient to show that the synergistic advantage of low-level fusion not only exists, but can even be significant, and that our setup is one way to access and employ this advantage in a safety-critical application. In our current paper we aim to show the existence of the fusion benefits, but not the universality. Performing quick comparisons with other popular or well-performing network designs wouldn't add to the argument as any performance difference could never be clearly attributed to our fusion method exclusively, since the choice of architecture itself is also a defining factor. To argue for universality we would have to custom-fit our fusion onto several popular base designs and perform similar ablation studies reevaluating them under sparse conditions (this could be a topic for future research). A simple theoretical argument for universality can be formulated by noting that our fusion relies on a lossless augmentation of the input (and corresponding expansion of the base network), thus given enough training time and computational capacity, every fusioned neural design should be able to perform at least as good as the baseline design (by effectively regressing to the baseline in the worst case).

According to our results, we have shown that more reliable detection of distant targets that are characterized by very sparse LiDAR measurements is possible. In conclusion, we have seen that introducing lower levels of fusion into existing perception architectures for autonomous vehicles can be beneficial in accomplishing specific safety-critical tasks like the detection of distant or heavily obscured objects. We intend to explore further possibilities for improving existing designs and devising new, raw-fusion specific architectures in the future in order to establish the extent of possible benefits of leveraging inter-sensor synergies.

Funding: The project has been supported by the European Union, co-financed by the European Social Fund EFOP-3.6.2-16-2017-00002.

References

1. Geiger, Andreas, Philip Lenz, Christoph Stiller and Raquel Urtasun. 2013. *Vision meets Robotics: The KITTI Dataset*. Technical Report October. http://www.cvlibs.net/datasets/kitti.
2. Geiger, Andreas, Philip Lenz and Raquel Urtasun. 2012. "Are we ready for autonomous driving? The KITTI vision benchmark suite." In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, jun, 3354–3361. IEEE.
3. Hall, David L. and James Llinas. 1997. "An introduction to multisensor data fusion." *Proceedings of the IEEE* 85 (1): 6–23.
4. He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 2016. "Deep residual learning for image recognition." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem: 770–778. http://arxiv.org/abs/1512.03385.
5. Kim, Seungki, Hyunkyu Kim, Wonseok Yoo and Kunsoo Huh. 2016. "Sensor Fusion Algorithm Design in Detecting Vehicles Using Laser Scanner and Stereo Vision." *IEEE Transactions on Intelligent Transportation Systems* 17 (4): 1072–1084. http://ieeexplore.ieee.org/document/7322252/.
6. Kovacs, L., L. Lindenmaier, H. Nemeth, V. Tihanyi and A. Zarandy. 2018. "Performance Evaluation of a Track to Track Sensor Fusion Algorithm." In *CNNA 2018; The 16th International Workshop on Cellular Nanoscale Networks and their Applications*, aug, 1–2.
7. Ku, Jason, Melissa Mozifian, Jungwook Lee, Ali Harakeh and Steven Waslander. 2017. "Joint 3D Proposal Generation and Object Detection from View Aggregation." http://arxiv.org/abs/1712.02294.
8. Lang, Alex H., Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang and Oscar Beijbom. 2018. "PointPillars: Fast Encoders for Object Detection from Point Clouds." http://arxiv.org/abs/1812.05784.
9. Liang, Ming, Bin Yang, Shenlong Wang and Raquel Urtasun. 2018. "Deep Continuous Fusion for Multi-Sensor 3D Object Detection." In *Cvpr 2018*, 16. http://openaccess.thecvf.com/content_ECCV_2018/papers/Ming_Liang_Deep_Continuous_Fusion_ECCV_2018_paper.pdf.
10. Lin, Tsung-yi, Piotr Doll, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, Facebook Ai and Cornell Tech. 2017. *(FPN) Feature Pyramid Networks for Object Detection*. Technical Report. http://openaccess.thecvf.com/content_cvpr_2017/papers/Lin_Feature_Pyramid_Networks_CVPR_2017_paper.pdf.
11. Lin, Tsung Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C. Lawrence Zitnick. 2014. "Microsoft COCO: Common objects in context." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8693 LNCS (PART 5): 740–755. http://arxiv.org/abs/1405.0312.
12. Qi, Charles R., Wei Liu, Chenxia Wu, Hao Su and Leonidas J. Guibas. 2017. "Frustum PointNets for 3D Object Detection from RGB-D Data." http://arxiv.org/abs/1711.08488.
13. Qi, Charles R., Hao Su, Kaichun Mo and Leonidas J. Guibas. 2016. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." http://arxiv.org/abs/1612.00593.
14. Redmon, Joseph and Ali Farhadi. 2018. *YOLOv3: An Incremental Improvement*. Technical Report. http://arxiv.org/abs/1804.02767.
15. Shi, Shaoshuai, Xiaogang Wang and Hongsheng Li. 2018. "PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud." http://arxiv.org/abs/1812.04244.
16. Wang, Zhixin and Kui Jia. 2019. "Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal

3D Object Detection." http://arxiv.org/abs/1903.01864.
17. Xu, Danfei, Dragomir Anguelov and Ashesh Jain. 2017. "PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation." 10. http://arxiv.org/abs/1711.10871.
18. Zhou, Yin and Oncel Tuzel. 2018. "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4490–4499. http://arxiv.org/abs/1711.06396.

Bionotes

András Rövid
Department of Automotive Technologies at the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary
andras.rovid@gjt.bme.hu

Mr. András Rövid graduated in 2001 from the Faculty of Electrical Engineering and Informatics at Technical University of Kosice. He earned his PhD degree in Transportation Sciences in 2005 from Budapest University of Technology and Economics (BUTE). He is currently senior research fellow at the Department of Automotive Technologies, BUTE where he is leading the Environment Perception Team. His main interest include image processing, 3D machine vision, environment perception for autonomous driving.

Viktor Remeli
Department of Automotive Technologies at the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary
viktor.remeli@gjt.bme.hu

Mr. Viktor Remeli graduated in 2015 from the Faculty of Information and Communication Technology at the University of Malta. He is currently research assistant at the Department of Automotive Technologies, BUTE where he also conducts his PhD studies. His research focuses on deep learning based environment perception methods and their verification.

Zsolt Szalay
Department of Automotive Technologies at the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary
zsolt.szalay@gjt.bme.hu

Mr. Zsolt Szalay, Ph.D. Head of the Department of Automotive Technologies at the Budapest University of Technology and Economics, Head of Research and Innovation at ZalaZONE, the unique Hungarian automotive proving ground for connected and automated vehicles. Founder of Inventure Automotive, a global supplier of CAN data retrieving solutions and Leader of the BME Automated Drive Laboratory. Graduated as M.Sc. both in Electrical Engineering and in Business Administration, obtained his Ph.D. degree in 2002. Member of the Hungarian Academy of Engineering since 2009. His research focus is highly automated vehicles, especially the testing and validation processes of CCAM and CAV technologies.