Realtime 3D Object Detection for Automated Driving Using Stereo Vision and Semantic Information

Hendrik Königshof[1], Niels Ole Salscheider[1] and Christoph Stiller[2]

***Abstract*— We propose a 3D object detection and pose estimation method for automated driving using stereo images. In contrast to existing stereo-based approaches, we focus not only on cars, but on all types of road users and can ensure real-time capability through GPU implementation of the entire processing chain. These are essential conditions to exploit an algorithm for highly automated driving. Semantic information is provided by a deep convolutional neural network and used together with disparity and geometric constraints to recover accurate 3D bounding boxes. Experiments on the challenging KITTI 3D object detection benchmark show results that are within the range of the best image-based algorithms, while the runtime is only about a fifth. This makes our algorithm the first real-time image-based approach on KITTI.**

I. Introduction

Estimation of the full motion state of all other dynamic objects is an essential information that enables fully automated driving. Because of the accurate depth information, currently, most of the 3D object detection methods heavily rely on LiDAR data. But depending on the exact model of the LiDAR sensor there are several disadvantages compared to a stereo camera regarding especially higher costs, but also shorter perception range and sparser information. Secondly, over-reliance on a single sensor is an inherent safety risk and therefore it is advantageous to have a second sensor available for detection of objects.

A stereo camera provides disparity images to detect, localize and reconstruct arbitrarily shaped objects in the scene. With semantic information obtained by a CNN the disparity based clustering can be improved and the type of the object can be established. This allows a complete reconstruction even for partially occluded or truncated objects by using a class-specific shape prior.

In this work we present a real-time capable stereo-based 3D object detection approach for all kinds of road users. Due to the estimation of a confidence score per object, these detections can easily fused with other sensors for object detection as LiDAR or RADAR.

The paper is organized as follows: The next section presents related work and distinguishes it from our work. Section III gives a coarse overview of the method before the object detection is discussed in more detail in Section IV. Some results and the evaluation are shown in Section V before the paper is concluded by a summary and an outlook.

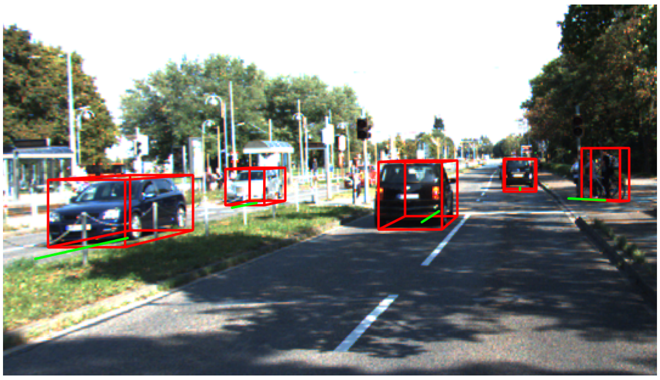


Fig. 1: 3D bounding boxes of objects projected into the input image (red) with corresponding orientation estimations (green).

II. Related Work

Robust environment perception with different sensor data is a well-studied problem. Here we give a brief overview about the related work that has been done on 3D object detection with LiDAR point clouds and images.

As LiDAR systems provide a high geometric accuracy for 3D world points, most of the approaches use a combination of both sensors by generating object proposals from RGB images and then estimating precisely segmented bounding boxes from the LiDAR point cloud. Du et al. [1] propose a flexible 3D vehicle detection pipeline to fuse the output of any 2D detection network with a 3D point cloud. Ku et al. [2] use LiDAR pointclouds and RGB images to generate features that are shared by a region proposal network and a second stage detector network for accurate oriented 3D bounding boxes. In the work of Schlosser et al. [3] they directly fuse LiDAR with an RGB image by converting the point cloud into an HHA map (horizontal disparity, height above ground, angle) and process the resulting six-channel RGB-HHA image with a CNN. Some methods [4] include the bird's eye view of the point cloud as additional input, because it has no projective loss as compared to the depth map and thus 3D proposal boxes can be generated directly. Liang et al. [5] project the image features into the bird's eye view and use continuous convolutions to fuse image and LiDAR feature maps at different levels of resolution.

Instead of generating proposals from RGB images or projecting the point cloud to bird's eye view or voxels, Shi et al. [6] directly generate 3D proposals from LiDAR point clouds in a bottom-up manner via segmenting the point cloud into

[1] Intelligent Systems and Production Engineering, FZI Research Center for Information Technology, 76131 Karlsruhe, Germany {koenigshof,salscheider}@fzi.de

[2] Institute of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany stiller@kit.edu

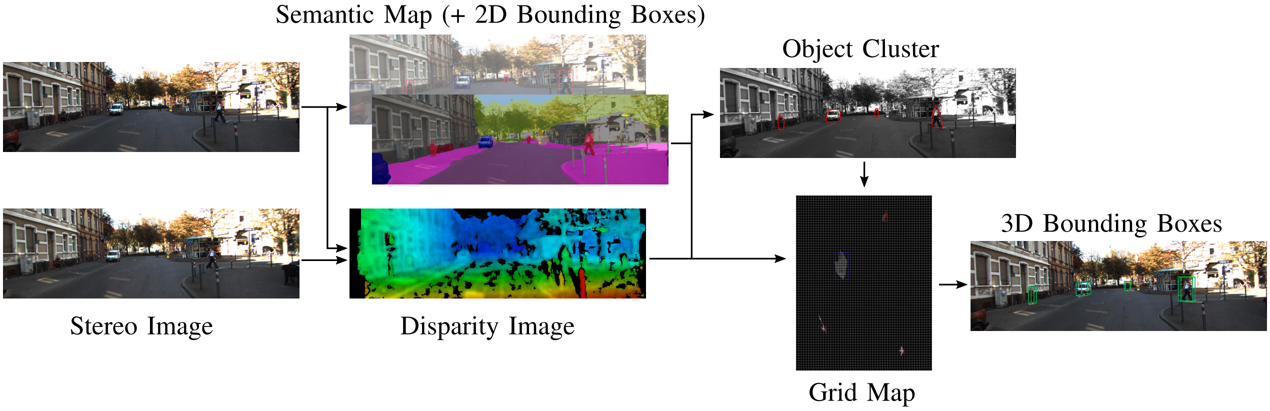


Fig. 2: **The proposed pipeline for our stereo-based 3D object detection:** The left image is used to generate a semantic map and optional bounding box suggestions, together with the right image disparities are calculated. These are clustered and projected into a grid map with elevation information, which is then used to estimate the 3D bounding boxes.

foreground points and background. In the approach of Li et al. [7] the 3D point cloud data is projected in a 2D point map and a single 2D end-to-end fully convolutional network is used to identify the vehicle bounding boxes. Engelcke et al. [8] perform object detection in point clouds with CNNs constructed from sparse convolutional layers based on a voting mechanism.

Other approaches [9] try to estimate complete 3D bounding boxes by using geometric constraints provided by a 2D bounding box from monocular images. Similarly in Deep-MANTA [10], the vehicle orientation, size and location of key points on the car are estimated by a CNN and a 2D/3D shape matching algorithm is applied to estimate the 3D pose of the vehicle. Xiang et al. [11] detect 3D Voxel Patterns, that capture the key properties of objects including appearance, 3D shape, viewpoint, occlusion and truncation in the 2D image. Chen et al. [12] propose an energy minimization approach that samples candidate bounding boxes in the 3D space by assuming a prior on the ground-plane and then score each candidate box projected to the image plane by exploiting multiple features like semantic or shape. A drawback of all these approaches is that they are very sensitive to the assumptions made.

Surprisingly, there are only a few works that use stereo vision to recognize 3D objects. Most of them take advantage of a two-stream CNN where the RGB channel and either the disparity map [13] or an HHA image [14] go through two separate CNN branches and are concatenated before the prediction layers, where class labels, bounding box coordinates and the object orientation is predicted jointly using a multi-task loss. According to the KITTI evaluation, the most promising methods currently available are Stereo R-CNN [15], which detects and associates objects simultaneously in the left and right image and then recovers the accurate 3D bounding box by a region-based photometric alignment. And secondly, a so called "Pseudo-LiDAR" approach [16] converting image-based depth maps to a LiDAR representation and apply different existing LiDAR-based detection algorithms.

In contrast to our method, none of these approaches is able to provide an object list of all relevant other road users in real time (total latency less than 100 ms). However, this is an essential criterion for highly automated driving.

III. System Overview

Figure 2 summarizes our system: As input we need a stereo image pair, either grayscale or color. The left image is used to obtain pixelwise semantic information using a CNN. Additionally a 2D bounding box detection can be performed with this network. This changes the results of our approach only slightly, but provides a better parallelization of the clustering and thus a lower runtime. In parallel to receiving the semantic information, a block matching algorithm is used to calculate the disparities between left and right image. Subsequently, a clustering of the disparities is performed using Connected Component Labeling to obtain object proposals. A lower threshold is used if pixels belong to the same semantic class and have a higher confidence. For all cluster points found in image domain the position in world coordinates is computed and projected into a Grid Map in the xz-plane with height information representing the y-coordinate. After filtering outliers, the orientation and dimensions of the object are optimized by using class specific shape priors. Finally the 3D bounding box can be calculated.

IV. Object Detection

A. *Semantic Classification and Bounding Box Detections*

For pixelwise semantic segmentation and object detection in images, we use a Convolutional Neural Network. The backbone is an encoder based on ResNet-38 [17]. Due to feeding the output of the backbone into two heads, this network solves both tasks simultaneously. The first head decodes the backbone output to a semantic segmentation map with the original image resolution. The second head performs bounding box detection and regression. It uses a proposal-free approach that takes ideas from SSD [18] and RetinaNet [19]. The detailed architecture of our network is described in [20].

B. Disparity Estimation

For disparity estimation, a block matching algorithm based on [21] is used. This algorithm takes advantage of a slanted planes approach and is still among the fastest block matching approaches due to the GPU implementation, while providing quite good results. By combining the following metrics obtained by the block matcher we are able to calculate a confidence metric which can later be used for clustering and estimation of an existence probability per object. This is essential in order to be able to fuse the resulting stereo-based object detections with those of other sensors. The *Peak-Ratio*

$$C_{\mathrm{PKR}}(u,v) = \frac{c_{\mathrm{min2}}(u,v)}{c_{\mathrm{min}}(u,v)} \tag{1}$$

compares the local minimum of the matching cost curve c_{min} with the second lowest costs c_{min2}. A high value represents a distinct minimum and thus a higher certainty that the determined disparity is correct. The *Left-Right-Consistency*

$$C_{\mathrm{LRC}}(u,v) = \left| d^{\mathrm{l}} - d^{\mathrm{r}}(u - d^{\mathrm{l}}, v) \right| \tag{2}$$

describes the consistency between both disparity maps. Here low costs ensure coherence of the left-view disparity map d^{l} and the projected right-view disparity map d^{r}.

C. Clustering

The computed disparities are now clustered into groups of similar values. For this purpose, an undirected graph is built up where each pixel is considered as node and adjacent nodes that lie within an semantic and confidence dependent adaptable threshold are connected by edges. From this graph, the connected components are computed using a DFS-based approach. Each connected component is a set of vertices in this graph that are all reachable from each other. This step is also parallelized by dividing the image into individual blocks and then performing the Connected Component Labelling (CCL) in parallel within each block. Afterwards, clusters that go beyond the boundaries of the blocks must be combined. To do this, all block boundaries are sequentially checked for equivalent labels and afterwards the labels in each block are updated in parallel. However, the fact that the second step must be executed single-threaded results in a synchronization barrier before and after this step and due to that to runtime losses. As already mentioned above, these can be avoided by using 2D bounding boxes as additional input. In this case, the CCL is performed in parallel in all bounding boxes. Then only the few adjacent or overlapping bounding boxes are searched for clusters that need to be combined. The latter step is only used to be robust against wrong bounding box proposals. Furthermore, in all image areas without bounding boxes, a coarse clustering is applied in order to prevent to depend too strongly on the bounding box proposals and detect also objects that were not recognized by the CNN.

D. 3D Bounding Box Estimation

The resulting clusters are used as object proposals. All points of each cluster are projected into a Grid Map with resolution $[res_{\mathrm{x}}, res_{\mathrm{z}}]$ in x-z-plane by using the position of the pixel in the image $[u, v]$, disparity d, baseline b and the focal length f:

$$x = \frac{(u - u_0) \cdot b}{d \cdot res_{\mathrm{x}}} \tag{3}$$

$$z = \frac{f \cdot b}{d \cdot res_{\mathrm{z}}} \tag{4}$$

In addition, a minimum and maximum height per grid cell is calculated from all the

$$y_{\mathrm{i}} = \frac{(v_{\mathrm{i}} - v_0) \cdot b}{d(u_{\mathrm{i}}, v_{\mathrm{i}})} \tag{5}$$

of points lying in that cell.

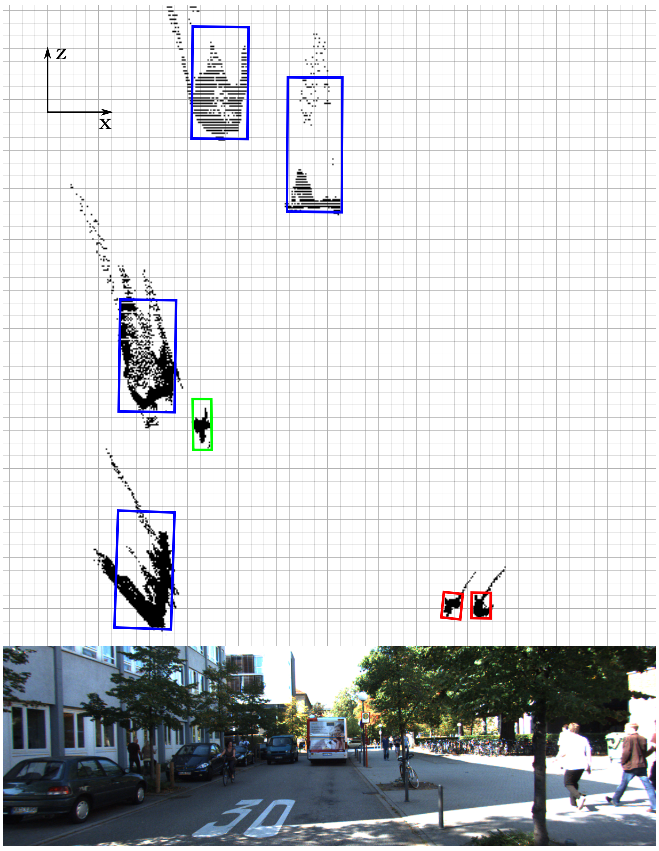
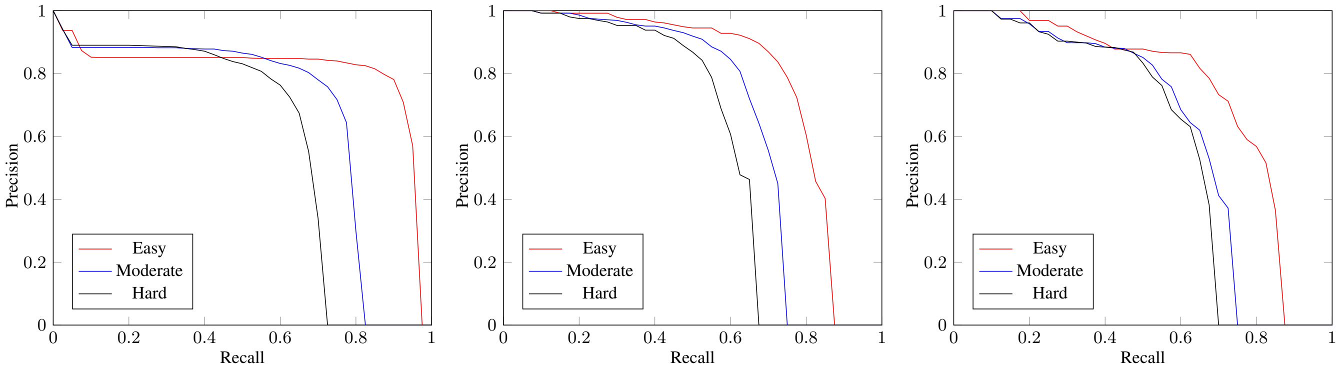


Fig. 3: Left camera image and a part of the resulting grid map including occupied cells (black) and reconstructed bounding rects for cars (blue), pedestrians (red) and cyclists (green), the displayed grid has a cell width and height of $0.5\,\mathrm{m}$ each, the actual resolution of the grid map used for reconstruction is $0.1\,\mathrm{m}$ in x- and z-direction.

Afterwards, a morphological opening (erosion followed by dilation) is done for filtering of outliers. A bounding rectangle is estimated per object around the remaining occupied grid map cells. Due to the mostly lower mounting position of cameras compared to LiDAR sensors, it may happen,



(a) Cars

(b) Pedestrians

(c) Cyclists

Fig. 4: Precision-recall curves for 2D detections on our validation set.

depending on the observation angle, that the complete object cannot be reconstructed. For example, in the case of another vehicle directly in front of our vehicle, where you would only be able to perceive the rear side.

Therefore, in such cases we use a "shape prior", which has different sizes (and variances) depending on the previously determined semantics of the object. Additionally we calculate the convex hull of all occupied grid map cells per object. Using this convex hull we optimize the orientation and exact dimension of the object, which maximizes the number of inlier grid map cells. It is assumed that additional unobserved cells of the object can only be located behind occupied cells in the direction of the observation angle. An exemplary grid map resulting from these steps including the corresponding left camera image is depicted in Figure 3.

A confidence score per object is calculated from all confidence values per pixel and the size of the object cluster in image domain. Since the area of an object in the image decreases quadratically with its distance in world coordinates, the quadratically growing depth error which occurs using stereo cameras is implicitly considered.

V. Results and Evaluation

A. Implementation Details

We evaluate our approach on the KITTI object detection benchmark [22], which has 7481 training images with available ground truth labels and 7518 testing images. To enable the estimation of a disparity image, corresponding rectified right images are also provided. The benchmark contains three different object classes: Car, Pedestrian and Cyclist. For each class, the evaluation is divided into easy, moderate and hard groups based on their visibilities. Even though the evaluation only focuses on these three classes, our approach provides detections for all semantic classes listed in TABLE I. For the optimization of the bounding box orientation as described in Section IV-D, the mean values and standard deviations of height h, width w and length l provided in this table are used as shape prior. The evaluation on KITTI is split into 2D, 3D and bird's eye view evaluation. For all three the average precision is calculated by using an Intersection over Union (IoU) threshold of 0.7 for cars and an IoU of 0.5 for cyclists and pedestrians.

Class	**Occurrence**	μ_{dim}	σ_{dim}
	%	h, w, l $[m]$	h, w, l $[m]$
Car	70.8	1.53, 1.63, 3.88	0.14, 0.10, 0.43
Pedestrian	11.1	1.76, 0.66, 0.84	0.11, 0.14, 0.23
Van	7.2	2.21, 1.90, 5.08	0.32, 0.17, 0.83
Cyclist	4.0	1.74, 0.60, 1.76	0.09, 0.12, 0.18
Truck	2.7	3.25, 2.59, 10.11	0.45, 0.22, 2.86
Misc	2.4	1.91, 1.51, 3.57	0.81, 0.67, 2.86
Tram	1.3	3.53, 2.54, 16.09	0.18, 0.22, 7.86
Sitting person	0.6	1.27, 0.59, 0.80	0.11, 0.08, 0.22

TABLE I: Semantic Classes available in the KITTI Object Detection Evaluation 2017. Occurrences, mean values and standard deviations are shown here for the training data set. In the evaluation, vans are not considered as false positives for car and sitting persons are not considered as false positive for pedestrians due to their similarity in appearance.

B. 2D Bounding Box Evaluation

We train our neural network for object detection and semantic segmentation on the Cityscapes dataset [23] and the KITTI dataset [22]. From the Cityscapes dataset we use both the 5000 training images with fine annotations and the 20 000 training images with coarse annotations. From the KITTI training dataset we randomly selected 5930 images for training. The other images are used for our validation. Even if, as mentioned in section IV-C, the 2D bounding boxes only serve as an optional input for our clustering, the evaluation results of the detections in the image are briefly discussed here, as this also indicates the quality of the semantic map used. Figure 4 illustrates the precision-recall curves for all three object classes on KITTI. By training on Cityscapes and KITTI, the results cannot fully compete with the best 2D detection approaches on KITTI. But they deliver good results independent of the exact camera model, for example in our test vehicle *BerthaOne*, which are sufficient as bounding box proposals for our clustering. The collapse of precision for cars even with small recall values is mainly due to the often too large number of bounding boxes in lines of parked cars. However, these faulty bounding boxes are reliably filtered during clustering.

C. 3D Bounding Box Evaluation

Since an Intersection over Union of 0.7 is relatively difficult to achieve, especially for stereo-based approaches, Figure 5 depicts the accuracy depending on the IoU in the easy benchmark. There, it gets comparably high accuracies up to an IoU of about 50 %. For pedestrians and cyclists, who have much smaller dimensions, very good results are achieved up to an IoU of 30 %. Since the focus of our work is on the application for automated driving and in the case of cars an IoU of 50 % still means that the longitudinal and lateral position deviation is in the range of about 0.5 m, these results are sufficient to allow high-level scene understanding and trajectory planning for automated driving.
Furthermore it should be mentioned that experiments on our test vehicle have shown that with a higher resolution than KITTI's 0.4 MP significantly better results are achieved especially for distant and small objects such as pedestrians and cyclists.

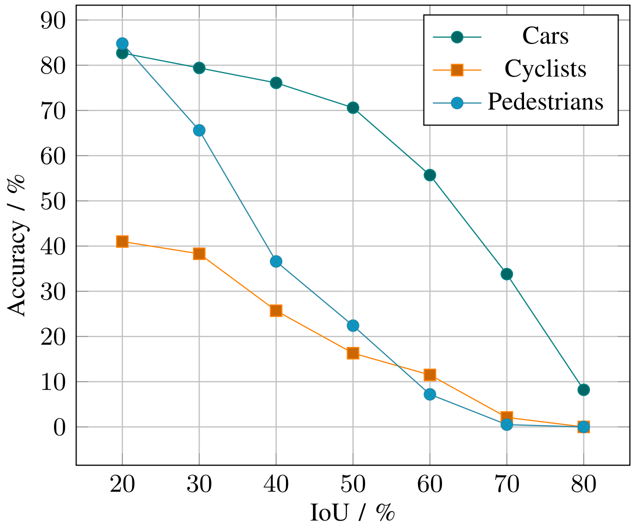


Fig. 5: Overall validation accuracy depending on the Intersection-over-Union (IoU) in the easy benchmark.

Nevertheless, TABLE II summarizes the quantitative evaluation results using the above mentioned benchmark metric of KITTI. Although the results on the test dataset may not fully match those on the validation dataset, they are within the range of the best stereo-based approaches [15] [16] while having a runtime of only one fifth. A comparison with LiDAR based algorithms should not be made anyway for several reasons. Firstly, it provides a higher geometric accuracy, which is in addition approximately constant over all distances. Secondly, the higher mounting position of the LiDAR compared to the cameras allows a better reconstruction of the depth of several objects. Last but not least, it should not be forgotten that even small errors in the given calibration between camera and LiDAR can cause 3D position errors, as KITTI's ground truth was labeled in the LiDAR point clouds.

Benchmark	**Easy**	**Moderate**	**Hard**
Car (2D Detection)	57.56 %	48.92 %	42.81 %
Car (3D Detection)	28.50 %	24.10 %	20.32 %
Car (Bird's Eye View)	59.32 %	49.48 %	43.16 %
Pedestrian (2D Detection)	44.54 %	32.01 %	31.50 %
Pedestrian (3D Detection)	4.27 %	4.25 %	4.26 %
Pedestrian (Bird's Eye View)	5.39 %	5.30 %	5.19 %
Cyclist (2D Detection)	23.73 %	17.66 %	11.94 %
Cyclist (3D Detection)	6.62 %	6.63 %	4.03 %
Cyclist (Bird's Eye View)	7.70 %	7.59 %	7.51 %

TABLE II: KITTI evaluation results on the test dataset.

Figure 6 presents the runtimes for the individual processing steps when using the bounding boxes as additional input for clustering or not. As already described in section IV-C, the differences in computation times result from the better parallelization capability when using the bounding boxes. The processing times are evaluated on an NVIDIA TITAN X GPU with 12 GB graphics memory. These are the mean times over the complete test dataset and can vary accordingly per image depending on the number and size of the objects to be detected.

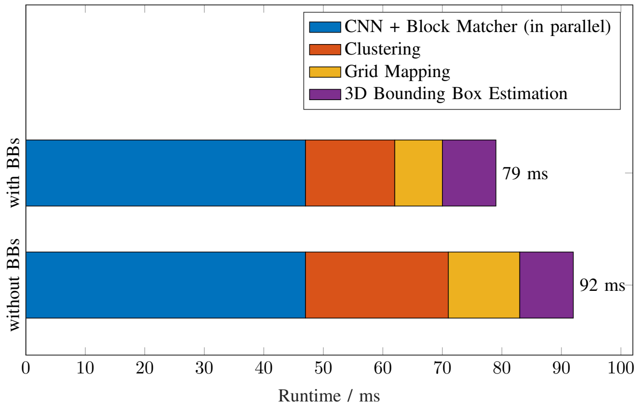


Fig. 6: Breakdown of runtimes of the processing steps involved in the proposed method. The semantic segmentation by the CNN and the stereo block matching can be executed in parallel and are therefore not listed here individually.

VI. Conclusion and Future Work

In this paper, we presented the first real-time capable stereo-based 3D object detection approach on KITTI. Also, it is the first method using images which focus not only on cars but on all types of road users. Due to the calculation of confidence values during the stereo block matching we are able to estimate a confidence score per object which is needed for the potential fusion with other object detecting sensors. Even if there is still a gap in the results to LiDAR-based approaches due to the worse geometric accuracy of stereo, this method offers a cost-effective alternative or a reliable backup in the event of a sensor failure.

Experiments on our test vehicle *BerthaOne* have also shown that using a resolution of 2 MP, even objects at a distance of 100 m can reliably be detected. Although with a slightly higher position uncertainty, but still within a sufficient range for high level scene understanding and trajectory planning. As a next step we want to use not only a single image pair for the detection of objects but several consecutive ones which will allow for a more robust reconstruction and tracking of the objects while still maintaining real-time capability.

References

[1] X. Du, M. H. Ang Jr, S. Karaman, and D. Rus, "A general pipeline for 3d detection of vehicles", *ArXiv preprint arXiv:1803.00387*, 2018.

[2] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation", in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1–8.

[3] J. Schlosser, C. K. Chow, and Z. Kira, "Fusing lidar and images for pedestrian detection using convolutional neural networks", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 2198–2205.

[4] 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*, 2017, pp. 1907–1915.

[5] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 641–656.

[6] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud", *ArXiv preprint arXiv:1812.04244*, 2018.

[7] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network", *ArXiv preprint arXiv:1608.07916*, 2016.

[8] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks", in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1355–1361.

[9] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7074–7082.

[10] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[11] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1903–1911.

[12] 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*, 2016, pp. 2147–2156.

[13] C. C. Pham and J. W. Jeon, "Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks", *Signal Processing: Image Communication*, vol. 53, pp. 110–122, 2017.

[14] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals using stereo imagery for accurate object class detection", *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1259–1272, 2018.

[15] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving", *ArXiv preprint arXiv:1902.09738*, 2019.

[16] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving", *ArXiv preprint arXiv:1812.07179*, 2018.

[17] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition", *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, *et al.*, "Ssd: single shot multibox detector.", in *European Conference on ComputerVision*, 2016.

[19] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection.", in *ICCV*, IEEE Computer Society, 2017.

[20] N. O. Salscheider, "Simultaneous object detection and semantic segmentation", *ArXiv preprint arXiv:1905.02285*, 2019.

[21] B. Ranft and T. Strauß, "Modeling arbitrarily oriented slanted planes for efficient stereo vision based on block matching", in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, IEEE, 2014, pp. 1941–1947.

[22] 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.

[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, *et al.*, "The cityscapes dataset for semantic urban scene understanding.", in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.