Dynamic Door Modeling for Monocular 3D Vehicle Detection

Thomas Barowski[1], Andre Brehme[1], Magdalena Szczot[1] and Sebastian Houben[2]

***Abstract*— The precise 3D localization of non-ego vehicles is a crucial task for the long-term goal of autonomous driving. In urban scenarios, where pedestrians frequently interact with vehicles, this task also requires a precise modeling of dynamic vehicle parts, e.g., doors. Current state-of-the-art computer vision algorithms are in fact able to estimate a vehicle pose but do not model doors by any means.**

To provide a solution solely based on a monocular camera, our proposed pipeline first performs a six degree-of-freedom pose estimation and then predicts the respective states of the vehicle doors. For both problems we utilize a perspective-n-point fitting method based on key points. To this end, we jointly detect the two required sets of correspondences for the vehicle body and the doors with a neural network. Since little insight is published for the application of key point based vehicle detection in the literature, we compare different implementations of the key point prediction module and investigate algorithm details, i.e., the role of a key point visibility analysis and two differing key point layouts. Results for the body estimation and the door detection with respect to these implementation details are presented on a proprietary dataset, in which we utilize an exact vehicle model to receive precise ground truth.

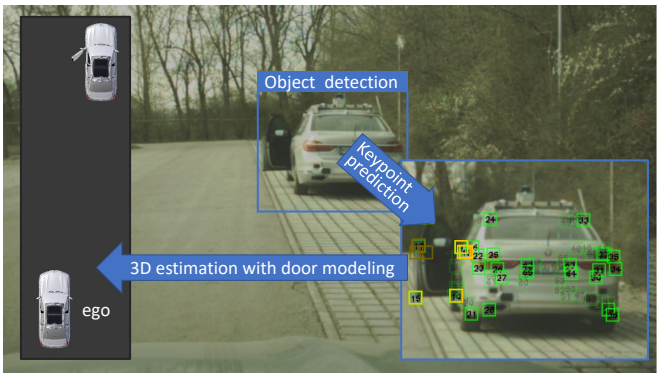

Fig. 1: The presented pipeline detects vehicles in a monocular RGB image and estimates two sets of key points that explicitly model the vehicle's doors (yellow) along with the body (green). A full pose estimation is done with two perspective-n-point fitting postprocessing blocks, both utilizing an abstract key point model. The outputs of the postprocessing blocks are combined to estimate the exact vehicle in 3D with respect to the door states.

I. Introduction

The precise detection and 3D localization of non-ego vehicles is an important aspect for the long-term goal of autonomous driving. From the various sensor techniques available, a solely camera-based solution is favorable since it offers cost-effectiveness, a small form factor and high information density compared to its alternatives, i.e., LIDAR or RADAR.

State-of-the-art object detection algorithms summarize vehicles as rigid bounding boxes, providing an efficient baseline representation but fail to grasp the full complexity of the vehicle's structure: The dynamic behavior of vehicle doors creates many situations – especially in urban scenarios – where this abstraction level results in a crash, e.g., colliding with an open vehicle door when solely relying on vision-based systems.

To present a solution for this problem, we extend a perspective-n-point (pnp) fitting pipeline to not only solve for the vehicle body pose but also the opening angles of all its doors in a two-stage design (Fig. 1). We train a neural network to jointly perform the tasks of object detection, key point prediction and key point visibility analysis. Existing works [1], [2], [3] present similar approaches but with varying implementations and without deeper insights to the key point detection performance. To provide these insights for the key point prediction problem, we perform an extensive analysis of three research aspects along with our vehicle door state evaluation: Firstly, we compare two different formulations of the key point detection problem – a heatmap approach and a regression-based strategy – resulting in different characteristics for the respective tasks. Secondly, two definitions for the required key point sets from literature are deployed as the vehicle model. Finally, the need for and influence of a joint key point visibility analysis is investigated.

We evaluate the door modeling and the key point prediction task on a proprietary dataset since the problem of vehicle door detection is not represented by any scientific, publicly available dataset. We provide an extended analysis of the general detection performance, the abstract key point models and high-precision door opening angle estimation.
In summary our contributions are:

- A multi-stage vehicle detection pipeline that explicitly considers dynamic vehicle parts in the returned 3D output,
- a comparison of three research aspects within the key point prediction task (prediction method, visibility, deployed key point model)
- and an evaluation of all mentioned characteristics on a proprietary dataset.

II. Related Work

This study is embedded in the domain of monocular single-shot computer vision algorithms for 3D vehicle de-

[1]Thomas Barowski and Magdalena Szczot are with BMW AG, Munich, Germany firstname.lastname@bmw.de

[2]Sebastian Houben is with the Institute for Neural Computation, University of Bochum, Germany sebastian.houben@ini.rub.de

Manuscript 233 submitted to 2021 32nd IEEE Intelligent Vehicles Symposium (IV).
Received February 14, 2021.

tection. Although this specification comes with strict limitations, i.e., the constraint that 3D information cannot be estimated directly from a camera, algorithms designed in that domain contribute enormously to autonomous systems: State-of-the-art algorithms for vehicle or general object detection currently often require a direct 3D measurement, e.g., from LIDAR or RADAR. These sensors come with high production cost, limited resolution and perform orders of magnitudes worse in classification problems. This led to a strong focus on camera-LIDAR-fusion approaches [4], [5], [6], [7] which combine the laser's accurate depth estimation with descriptive features from a camera.

When solely relying on a camera, the problem of joint 3D size and pose estimation is often solved by an exploitation of the temporal dimension, e.g., structure-from-motion [8], or stereo camera setups [9], [10]. Both of the approaches require their own global pre-processing, which is susceptible to difficult lighting conditions and other perturbations. Since monocular single-shot solutions lack these pre-processing steps, they are conceptually preferable. A popular line of research tries to solve the aforementioned 3D estimation as a pure learning problem with the help of neural networks, which rely on the tremendous learning capacities of deep learning. However, recent works [11], [12], [13] show that for open world 3D vehicle estimation the variance of the data is yet too high. Therefore model-based pipelines that include predictions from neural networks are a good trade-off for the targeted problem. Due to legislation and large-scale industrialization, vehicles are restricted to explicit manufacturing models in which the configuration can be measured once. What is more, vehicles can be well classified into groups (e.g., *limousine*, *SUV*, *truck*, ...), in which members of the same group share a basic chassis structure and have similar dimensions. This is the reason why the model-based keypoint detector by Chabot et al. [1] yielded outstanding performance on open world 3D vehicle detection benchmarks [14] with a still manageable amount of models. Their work has a learning component that slightly modifies the dimensions of the 3D model based on the visual cues from a bounding box detector. In comparison to our paper, this model adaptation lends itself well to 3D size estimation but precludes an accurate and proper evaluation of the used key point approach. They annotated the key points semi-automatically with the help of a 3D bounding box scaling, which is estimated from LIDAR and therefore prone to error. Although the distance estimation is accurate, the key point positioning is often very poor with their labeling pipeline. In contrast, using an exact vehicle model allows us to remove this deep coupling and to focus exclusively on the key point detection performance.

In the popular dataset by Song et al. [3], the authors use key point fitting for ground truth generation. Their study shows that pnp fitting is well suited for precise pose estimation, yet adapting the 3D vehicle models to the actual car in the frame yields imprecise key point positioning.

The neural network component of our pipeline is based on convolutional bounding box detectors which are widely known and used in the community. In short, we categorize it as a two-stage detector [15], [16], [17] with a VGG [18] backbone, in contrast to single-stage detectors [19], [20]. Interested readers are referred to [21], [22] for a concise overview of the field.

The problem of vehicle door modeling is to the best of our knowledge not covered extensively in current research. In scientific benchmarks [14], [23], [3], [24], which have a strong influence on the community by means of the provided data, it is neither treated as a subtask nor is it provided explicitly in the bounding box labeling.

III. Method

The method section starts with a presentation of the vehicle door detection pipeline followed by two approaches for predicting key points with a neural network. In the end, the two deployed key point models are explained.

Vehicle door detection pipeline. The goal of our pipeline is to detect an arbitrary number of vehicles in a given monocular image and predict the respective three-dimensional poses and extents while explicitly considering the vehicle's door states. To this end, we design a neural network for joint object detection and key point prediction, a six degree-of-freedom body vehicle estimation block and the consecutive door modeling. Both of the two latter post-processing steps use model-based approaches which exploit the optimization of the reprojection error between static 2D-3D correspondences with a calibrated camera. The pipeline is visualized in figure 2.

The detection of vehicles in a given RGB image is done with a Faster-RCNN [15] based network. Descriptive features are extracted with a convolutional network [18] and then passed to a region-proposal-network (RPN) that provides promising regions of interest by evaluating sample positions from an anchor-based grid structure. For each anchor point various bounding boxes are tested which are based on configurable sizes and aspect ratios. Interested readers are referred to [15], [16], [17] for details. The 300 best region proposals are passed on to the second stage of the network. The classification head verifies if the passed region contains an object of a known class (via softmax classification) or should be rejected as background. Additionally, the given bounding box is refined with values from a regression head. To utilize perspective-n-point fitting we extend this second stage head with a key point prediction module that outputs coordinate pairs (x_p,y_p) in the image plane for two sets of key points $p_1^{body}, ..., p_N^{body}$ and $p_1^{door,1}, ..., p_M^{door,1}, p_1^{door,2}, ..., p_M^{door,4}$: one for the body of the vehicle (*green*) and the other for modeling the doors (*yellow*). The amount and exact positions of these key point deployments are described in the last section of this chapter.

The two key point sets are passed to the respective post-processing blocks which use model-based approaches to estimate three-dimensional outputs. This means that the size

Manuscript 233 submitted to 2021 32nd IEEE Intelligent Vehicles Symposium (IV).
Received February 14, 2021.

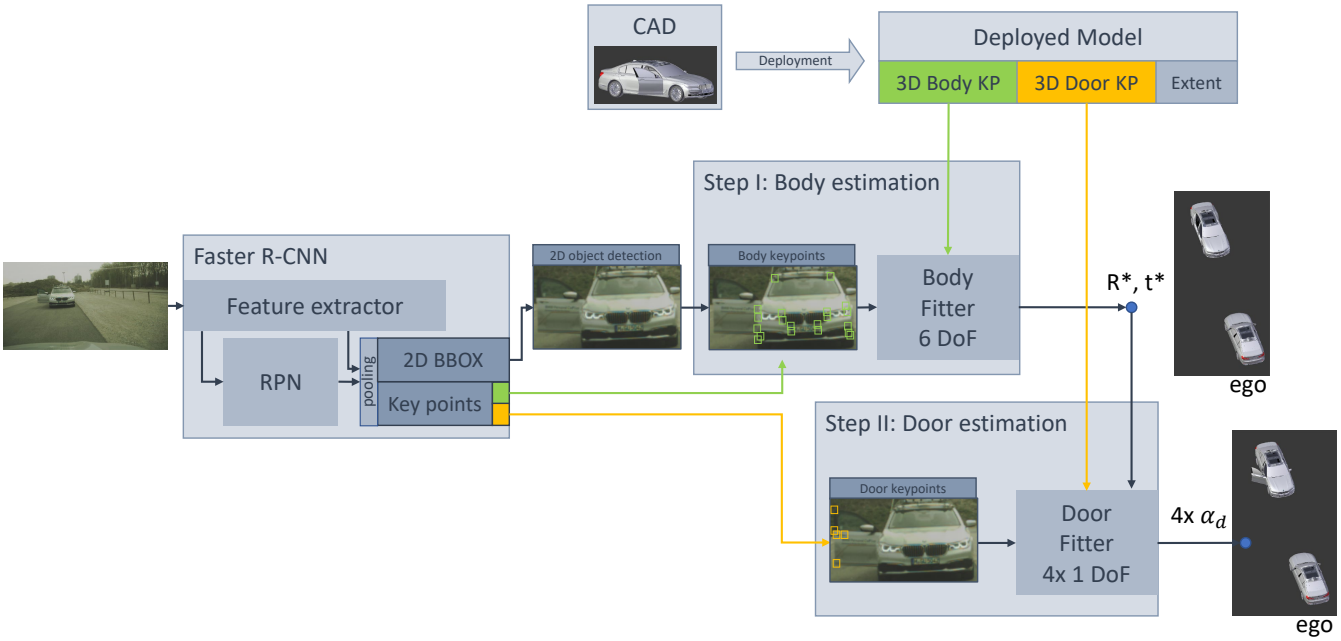

Fig. 2: Our multi-stage pipeline for joint estimation of the vehicle body pose and dynamic vehicle parts. A Faster R-CNN [15] based network predicts bounding boxes with two additional sets of key points that model the body (*green*) and dynamic parts (*yellow*). The first set is used to estimate the six degree-of-freedom vehicle pose which is then fixed to estimate the vehicle door status by solving a simplified optimization problem. For both steps an abstract 3D key point model is required which we obtain once from a high precision CAD model. As a combination of the post-processing blocks a six degree-of-freedom vehicle pose and four states for the vehicle doors are returned.

estimation is removed from the problem by using additional knowledge. In our case an exact CAD model of the vehicle to detect is utilized: the corresponding key points are annotated in 3D space and stored as a deployed model along with the vehicle's physical extent. Given a calibrated camera matrix K, perspective-n-point fitting is applied, which optimizes the reprojection error e_{proj} between the detected 2D key points $p_k^{body}, k = 1, ..., N$ and the projected coordinates of the corresponding 3D key points P_k^{body} over all possible object poses denoted by the rotation matrix R and the camera position t in 3D (eqn. 1-3). Efficient implementations like ePnP [25] solve this problem within a few milliseconds given at least four corresponding key point pairs. In addition to the coordinate prediction a visibility classification is done for each key point by the respective key point detection. The ground truth data for this visibility head is determined by raytracing with the CAD model similar to [1]. The quality of this visibility filtering and the influence on the estimation of 3D values will be evaluated in our experiments. We minimize

$$\arg\min_{R,t} \; e_{proj}(R,t) \tag{1}$$

$$e_{proj}(R,t) = \frac{1}{N}\sum_{k=1}^{N}\left(p_k^{body} - p_{proj,k}(R,t)\right)^2 \tag{2}$$

$$p_{proj,k}(R,t) = \pi\left(K \cdot R \cdot \left(P_k^{body} - t\right)\right), \tag{3}$$

where $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2, \pi(x,y,z) = (\frac{x}{z}, \frac{y}{z})$ performs the projection within the camera frame. The underlying assumption of a rigid body – a strict requirement of the perspective-n-point fitting – is the reason why the presented post-processing follows a two-stage design: Solving for both the six degree-of-freedom pose and the four degree-of-freedom door states in a single optimization problem leads to multiple convergence points and is therefore not considered. Instead, we first estimate the pose of the vehicle's body R^*, t^* with the key points from P_k^{body} and then utilize this pose to estimate the respective door states. By this means, the optimization problem of the doors can be reduced to four independent single input problems, i.e., the respective rotation α_d around a fixed and known rotation axis for each door $d = 1, ..., 4$ (Eq. (4)-(6)). Due to this simplification, a single key point would be sufficient to solve the optimization problem:

$$\arg\min_{\alpha_d} \; e_{proj,d}^{door}(\alpha_d) \tag{4}$$

$$e_{proj,d}^{door}(\alpha_d) = \frac{1}{M}\sum_{k=1}^{M}\left(p_k^{door,d} - p_{proj,k}(\alpha_d)\right)^2 \tag{5}$$

$$p_{proj,k}(\alpha_d) = \pi\left(K \cdot R(\alpha_d) \cdot R^* \cdot \left(P_k^{door,d} - t^*\right)\right) \tag{6}$$

where $P_k^{door,d}, k = 1, ..., M$ are the d-th door's 3D key points. Combining both results, the pipeline returns a six degree-of-freedom pose, the body's extent from the stored abstract model and an opening state for each door in the interval $[0; 1]$. With the knowledge from the model this opening state can be converted into an exact 3D measurement or an all-embracing bounding box if required.

Manuscript 233 submitted to 2021 32nd IEEE Intelligent Vehicles Symposium (IV).
Received February 14, 2021.

Key point prediction methods. The neural network can be configured with two different modules for the key point prediction: a regression approach [1] and a heatmap implementation [2]. Both modules predict a set of coordinate pairs (x_p,y_p) within the fine-tuned bounding box output for an arbitrary number of key points.

The first implementation [1] defines the key point coordinate estimation as a *regression* problem where the variables x_p and y_p are directly estimated within the possible bounding box range of $[0; 1]$ for each key point $p \in P$. The final amount of regression outputs is therefore $2 \cdot (N + 4M)$. During training the smooth L1 loss is applied to train the network. This way of modeling has the structural disadvantage that a key point will always be predicted to a point in the interval even if it is not visible since the output does not hold a non-visible state. Therefore, if the visibility of a key point shall be considered, an additional output is required which solely focusses on this task. The trivial implementation is the definition of $N + 4M$ additional binary classification problems with softmax loss. For each key point p the network will predict a binary filter value v_p to return only key points where $v_p = 1$.

The alternative *heatmap* implementation [2] solves the key point prediction task as a classification problem. Each bounding box is divided into a $K \times J$ grid where each cell represents a possible key point location. During training, a softmax loss is combined with a one-hot encoded label for closest key point grid cell. This is done independently for all $P = N + 4M$ key points, resulting in a $P \times K \times J$ output layer. In our experiments we set $K = J = 56$, as proposed in the original paper. The resulting grid indices are normalized on the grid size and scaled with the bounding box size to receive the final key point coordinates (x_p, y_p).

Deployed key point models. While the perspective-n-point fitting has a strict formalism, the resulting question where to place key points on a model and which amount leads to best performance is highly dependent on the application at hand. Related work [1], [3] has proposed two different abstractions: Firstly, the model called *MANTA* [1] defines a symmetrical structure of 36 key points as a coarse wireframe. Key points mostly capture the outlines of the chassis with an additional focus on the wheels (five key points each). There are no key points near the longitudinal axis and since there are no key points on the doors we had to add 18 key points on the outlines of dynamic parts, resulting in a final count of 54 key points. We call this model *MANTA+* in the experiments.

The second model *Apollocar* [3] already provides a fine-grained layout: 66 key points are labeled mainly on design elements (e.g., handles) and part transitions, where local gradients appear. This layout can be used as an abstract model in both tasks without further modification. The exact locations of the key points are depicted in figure 3.

Summary. We proposed a multi-step pipeline based on key point perspective-n-point fitting and introduced several research aspects within the approach: two competitive key point prediction methods, two key point arrangements as the abstract model and the influence of key point visibility analysis. All three topics will be addressed in the following experiments.

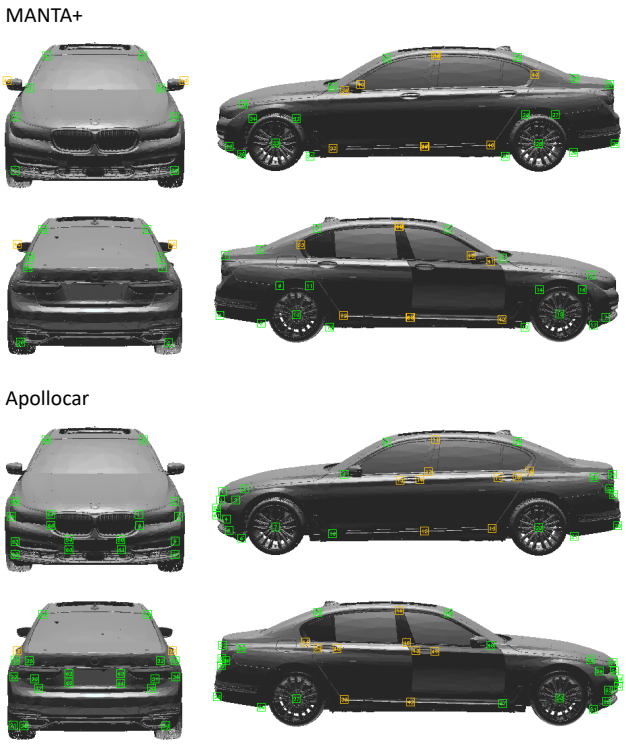

Fig. 3: The key point models *MANTA+* (*top*) [1] and *Apollocar* (*bottom*) [3] annotated on a rendering of our CAD vehicle model. Green key points are used to estimate the vehicle body while yellow key points are for the doors. The final deployed models hold the 3D coordinates of both key point sets, the physical extent of the vehicle and four transformation matrices to the rotation axes of the doors.

IV. Experiments

Our experiments analyze the performance of key point-based approaches for vehicle door modeling along with a six degree-of-freedom pose estimation. We start with a review of our proprietary dataset followed by framework implementation details. The utilized multi-stage pipeline does not allow for a straightforward analysis of the overall performance, thus, we present a stagewise evaluation concept: We briefly present image-plane related metrics before analyzing the performance of the body pose estimation followed by the final vehicle door detection metrics. Subsequently, we evaluate the coupling between the latter two stages. The names of all methods under examination have been introduced in section III.

Dataset. Although vehicle detection and pose estimation are very popular in scientific benchmarks ([14], [23], [24], [26]), the task of vehicle door detection was never proposed.

Manuscript 233 submitted to 2021 32nd IEEE Intelligent Vehicles Symposium (IV).
Received February 14, 2021.

This is in our opinion due to three reasons: A focus on LIDAR-based sensor setups, the difficult integration of doors into bounding box based detectors and the high requirements on labeling, especially in open world benchmarks. To overcome these problems we recorded a dataset with a single vehicle type, utilizing an exact production-level CAD model of our autonomous driving test vehicles. While other work [1], [3] focuses on a generalization of the key point models to detect key points for any type of vehicle, the restriction on a single but exact vehicle model allows for investigating the performance of the key point detection modules in a precise manner. While the annotated key points in related work [1], [3] only coarsely fits the wanted parts, we receive ground truth on real-world images within a precision of a few centimeters, both for the vehicle's body and doors.

The image data is recorded with a calibrated automotive camera and automatically labeled with an object pose estimated by a dGPS system. We manually fine-tune this object pose and label the door status with help of the projected CAD model. In the end, a frame is described by its image, the six degree-of-freedom vehicle pose, the dimensions of the vehicle model and four values between 0 and 1 to model the door status. Due to the high effort for labeling, the resulting dataset consists of 200 images, divided 175/10/15 into training, validation and testing dataset.

Implementation and network training. Our implementation is based on the tensorflow objection detection framework [22] which already implements a two-stage 2D object detection head and the mask-rcnn [2] heatmap approach. We extend the classification stage with the alternative key point prediction via regression. All models are trained for 800K steps with learning rate reduction by a factor of 10 at 500K and 700K steps. Hyperparameter optimization on the validation dataset showed that heatmap approaches require a higher initial learning rate of 1e-2 compared to the regression approach (1e-3).

2D object detection. Due to the multi-stage detector design the general object detection performance is evaluated first to prove a valid detection baseline for the subsequent tasks: All trained models have 100% recall and peform in the same orders of magnitude for **mAP** and **mIoU** with good results above 90% (Tab. I).

Vehicle body pose estimation. The performance of the body pose estimation is evaluated by mean Euclidean distances ($\Delta\mathbf{T}$) and the mean of the accumulated angular errors ($\Delta\mathbf{R}$). All regression models perform significantly better than the models with the heatmap approach as shown in Table II. In fact, without visibility analysis the translation error increases by one order of magnitude. The mean normalized key point localization error $\Delta\mathbf{K}$ underlines that the quality of the heatmap key point detection is worse by a factor of 2 to 3. Analyzing the influence of the visibilty classification leads to interesting insights: From the comparison of $\Delta\mathbf{K}$ with the localization error of visible key points only ($\Delta\mathbf{K}_{vis}$) we learn that visible key points are detected significantly more accurately. This is not caused by learning multiple tasks as a comparison with the groundtruth visibilities for the models without an output for visibilities ($\Delta\mathbf{K}_{vis}^{GT}$) shows. On the other hand, there is no performance increase in the pose estimation between the regression models with and without visibility filtering. Our intuition is that this is caused by the detection noise of the key point prediction, limiting the overall performance of the pnp fitter. Regarding the heatmap models, for which a precise prediction of the key points is difficult, the additional visibility analysis gives a strong benefit, reducing the translation error by a factor of 3 to 7. Therefore and because of the good classification results greater 95% in $\mathbf{P}_{vis}$, we recommend to implement the key point visibility prediction.

Approach	**Model**	**mAP** [%]	**mIoU** [%]
Regression	MANTA+	90.9	93.4
	Apollocar	90.9	93.8
	MANTA+ VIS	91.8	93.0
	Apollocar VIS	90.3	93.4
Heatmap	MANTA+	**94.7**	**94.1**
	Apollocar	91.8	93.7
	MANTA+ VIS	92.3	94.0
	Apollocar VIS	91.9	93.6

TABLE I: Performance of 2D vehicle detection via mean average precision (**mAP**) and mean intersection over union (**mIoU**) for the different key point prediction approaches and key point models.

Comparing the two deployed key point models, no performance difference during the pose estimation is noticed: The Apollocar model tends to perform slightly better, but not in a significant manner.

Vehicle door detection. Evaluating the dynamic door detection a similar main statement can be deduced: Regression models perform better than heatmap models. But the difference between both methods shrinks down to a factor of 1.5 to 2 when looking at the average door state error $\Delta\mathbf{E}_{dyn}$ in Table III. The discrete state analysis with two and three states (*open*, *half-open*, *closed*) in the columns $\mathbf{P}_{2S}$ and $\mathbf{P}_{3S}$ even leads to a smaller difference in performance. In a comparison of the key point detection quality between static ($\Delta\mathbf{K}^{body}$) and dynamic parts ($\Delta\mathbf{K}^{dyn}$) the static parts are detected better with a factor of nearly 2 for the favored regression models. The filtering of visible key points, presented in $\Delta\mathbf{K}_{vis}^{body}$ and $\Delta\mathbf{K}_{vis}^{body}$, again leads to key points with lower localization error and therefore better state classification precisions ($\mathbf{P}_{2S}$, $\mathbf{P}_{3S}$).

For the dynamic parts, the deployed key point model makes a difference: In the best case (Heatmap with VIS) the state classification precision benefits with an additional performance of 12 percentage points. But also all other models increase performance in all metrics. We conclude that the definition of key points on regions with strong gradients (e.g., handles, c.f., Fig. 3) are the main cause for this extra performance in the vehicle door modeling task.

Manuscript 233 submitted to 2021 32nd IEEE Intelligent Vehicles Symposium (IV).
Received February 14, 2021.

Approach	Model	$\Delta\mathbf{T}$ [m]	$\Delta\mathbf{R}$ [deg]	$\Delta\mathbf{K}$ [%]	$\Delta\mathbf{K}_{vis}$ [%]	$\Delta\mathbf{K}_{vis}^{GT}$ [%]	$\mathbf{P}_{vis}$ [%]
Regression	MANTA+	0.31	3.5	4.1		**1.5**	
	Apollocar	0.32	**3.3**	**3.7**		1.6	
	MANTA+ VIS	0.32	4.1	4.0	1.5	1.6	96.6
	Apollocar VIS	**0.29**	3.6	3.8	**1.4**	1.7	96.5
Heatmap	MANTA+	3.17	10.8	12.9		3.9	
	Apollocar	3.63	12.9	11.1		4.5	
	MANTA+ VIS	0.94	24.2	10.1	4.9	4.2	**96.4**
	Apollocar VIS	0.54	6.6	9.9	3.2	3.3	**96.4**

TABLE II: Performance of the six degree-of-freedom pose estimation task. Mean Euclidean distances are summed up to $\Delta\mathbf{T}$ while $\Delta\mathbf{R}$ represents the accumulated angular errors. The mean normalized key point localization errors for all key points, visible predicted key points and key points with visibility from ground truth are denoted by $\Delta\mathbf{K}$, $\Delta\mathbf{K}_{vis}$, $\Delta\mathbf{K}_{vis}^{GT}$ respectively. The key point localization error is normalized on the respective bounding box dimensions. Visibility classification precision is given in column $\mathbf{P}_{vis}$.

Approach	Model	$\Delta\mathbf{E}_{dyn}$	$\mathbf{P}_{2S}$ [%]	$\mathbf{P}_{3S}$ [%]	$\Delta\mathbf{K}^{body}$ [%]	$\Delta\mathbf{K}^{dyn}$ [%]	$\Delta\mathbf{K}_{vis}^{body}$ [%]	$\Delta\mathbf{K}_{vis}^{dyn}$ [%]
Regression	MANTA+	0.123	82.9	80.0	3.1	5.5		
	Apollocar	0.122	85.7	84.3	3.0	**4.8**		
	MANTA+ VIS	0.101	88.6	85.7	3.3	4.9	1.2	2.0
	Apollocar VIS	**0.086**	**91.4**	**88.5**	**2.8**	5.2	**1.1**	**1.8**
Heatmap	MANTA+	0.207	72.9	64.3	12.2	13.8		
	Apollocar	0.214	70.0	67.1	10.0	12.5		
	MANTA+ VIS	0.166	75.7	75.7	9.7	10.7	3.6	6.7
	Apollocar VIS	0.101	87.1	82.9	7.2	13.6	2.2	4.5

TABLE III: Performance of the estimation of the vehicle door states as mean state error $\mathbf{E}_{dyn}$ and two- and three-state classification (*open*, *half-open*, *closed*) precision $\mathbf{P}_{2S}$, $\mathbf{P}_{3S}$. Similar to Tab. II the mean normalized key point localization errors are denoted by $\Delta\mathbf{K}$ for the two sets (the body ^{body} and dynamic doors ^{dyn}) and whether or not to visibility analysis has been performed ($_{vis}$).

Body pose dependency analysis. Since the optimization of the vehicle state is a function of the estimated vehicle pose (Eqn. 4) we analyze the coupling between both modules in Table IV. Therefore, we estimate the dynamic door states using the vehicle pose ω^{GT} from ground truth and compare it to the results from the previous section.

Using the ground truth has little effect on the results of the regression models since they already perform well. However, for the heatmap models increases of up to +10% are measurable for the state classification precisions ($\mathbf{P}_{2S}$, $\mathbf{P}_{3S}$), indicating that for an accurate vehicle door estimation a robust vehicle pose is required.

Approach	Model	$\Delta\mathbf{E}_{dyn}^{GT}$	$\mathbf{P}_{2S}^{GT}$	$\mathbf{P}_{3S}^{GT}$
Regression	MANTA+	0.154	81.4	77.1
	Apollocar	0.140	81.4	81.4
	MANTA+ VIS	0.097	90.0	87.1
	Apollocar VIS	**0.083**	**92.9**	**88.6**
Heatmap	MANTA+	0.177	80.0	77.1
	Apollocar	0.171	80.0	77.1
	MANTA+ RVIS	0.146	82.9	80.0
	Apollocar RVIS	0.085	90.0	87.1

TABLE IV: Performance of the estimation of the vehicle door states using the ground truth pose ω^{GT} with the predicted door key points.

V. Conclusion

We presented an algorithm that explicitly models dynamic vehicle doors along with 2D detection and six degree-of-freedom pose estimation of vehicles. Our pipeline builds upon a model-based key point approach, in which both required sets of key points are predicted jointly. We implemented a heatmap and a regression key point prediction module from literature and compared their respective results. Also, the role of visibility analysis during fitting was investigated. For our evaluation we utilized a proprietary dataset and deployed two key point models which were also benchmarked.

Our results show that the regression approach provides significantly better results, not only in the estimation of vehicle poses but also in the state estimation of the vehicle doors. High classification rates for both two- and three-state door analysis proved the presented algorithm well-suited for the task of vehicle door modeling. The extensive analysis of key point detection errors w.r.t. visibility analysis showed the benefits of this filtering, especially since key points with visible cues are detected better. In the end, the best results for both tasks were achieved using the regression approach with visibility filtering and the deployed *Apollocar* vehicle key point model.

Although the presented algorithm is limited to a single model, it can be used as a reference system with multiple known vehicle models or be further generalized by learning approaches, as for example shown in [1].

References

[1] F. Chabot, M. Chaouch, J. Rabarisoa, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[3] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang, "Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5452–5462.

[4] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation,"

in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2018.

[5] 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*, 2018.

[6] 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.

[7] 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*, 2018.

[8] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] H. Königshof, N. O. Salscheider, and C. Stiller, "Realtime 3d object detection for automated driving using stereo vision and semantic information," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, 2019.

[11] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," 2019.

[12] E. Jörgensen, C. Zach, and F. Kahl, "Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss," *arXiv preprint arXiv:1906.08070*, 2019.

[13] Z. Qin, J. Wang, and Y. Lu, "Monogrnet: A geometric reasoning network for 3d object localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.

[16] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computing Research Repository (CoRR)*, vol. abs/1409.1556, 2014.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. of the European Conference on Computer Vision*, 2016.

[21] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, 2019.

[22] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[25] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, 2009.

[26] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," *arXiv: 1803.06184*, 2018.

Manuscript 233 submitted to 2021 32nd IEEE Intelligent Vehicles Symposium (IV).
Received February 14, 2021.