ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving

Xibin Song[1,2], Peng Wang[1,2], Dingfu Zhou[1,2], Rui Zhu[3], Chenye Guan[1,2],
Yuchao Dai[4], Hao Su[3], Hongdong Li[5,6] and Ruigang Yang[1,2]

[1]Baidu Research [2]National Engineering Laboratory of Deep Learning Technology and Application, China [3]University of California, San Diego [4] Northwestern Polytechnical University, Xi'an, China
[5] Australian National University, Australia [6]Australian Centre for Robotic Vision, Australia
{songxibin,wangpeng54,zhoudingfu,guanchenye,yangruigang}@baidu.com,
{rzhu,haosu}@eng.ucsd.edu, daiyuchao@gmail.com and hongdong.li@anu.edu.au

Abstract

Autonomous driving has attracted remarkable attention from both industry and academia. An important task is to estimate 3D properties (e.g. *translation, rotation and shape*) *of a moving or parked vehicle on the road. This task, while critical, is still under-researched in the computer vision community – partially owing to the lack of large scale and fully-annotated 3D car database suitable for autonomous driving research. In this paper, we contribute the first large-scale database suitable for 3D car instance understanding –* `ApolloCar3D`*. The dataset contains 5,277 driving images and over 60K car instances, where each car is fitted with an industry-grade 3D CAD model with absolute model size and semantically labelled keypoints. This dataset is above 20× larger than PASCAL3D+ [65] and KITTI [21], the current state-of-the-art. To enable efficient labelling in 3D, we build a pipeline by considering 2D-3D keypoint correspondences for a single instance and 3D relationship among multiple instances. Equipped with such dataset, we build various baseline algorithms with the state-of-the-art deep convolutional neural networks. Specifically, we first segment each car with a pre-trained Mask R-CNN [22], and then regress towards its 3D pose and shape based on a deformable 3D car model with or without using semantic keypoints. We show that using keypoints significantly improves fitting performance. Finally, we develop a new 3D metric jointly considering 3D pose and 3D shape, allowing for comprehensive evaluation and ablation study.*

1. Introduction

Understanding 3D properties of objects from an image, *i.e.* to recover objects' 3D pose and shape, is an important task of computer vision, as illustrated in Fig. 1. This task is also called "inverse-graphics" [27], solving which would enable a wide range of applications in vision and robotics, such as robot navigation [30], visual recognition [15], and human-robot interaction [2]. Among them, autonomous driving (AD) is a prominent topic which holds great potential in practical applications. Yet, in the context of AD the current leading technologies for 3D object understanding mostly rely on high-resolution LiDAR sensor [34], rather than regular camera or image sensors.

Figure 1: An example of our dataset, where (a) is the input color image, (b) illustrates the labeled 2D keypoints, (c) shows the 3D model fitting results with labeled 2D keypoints.

However, we argue that there are multitude drawbacks in using LiDAR, hindering its further up-taking. The most

Dataset	Image source	3D property	Car keypoints (#)	Image (#)	Average cars/image	Maximum cars/image	Car models #	Stereo
3DObject [52]	Control	complete 3D	No	350	1	1	10	No
EPFL Car [47]	Control	complete 3D	No	2000	1	1	20	No
PASCAL3D+ [65]	Natural	complete 3D	No	6704	1.19	14	10	No
ObjectNet3D [64]	Natural	complete 3D	Yes (14)	7345	1.75	2	10	No
KITTI [21]	Self-driving	3D bbox & ori.	No	7481	4.8	14	16	Yes
ApolloCar3D	Self-driving	industrial 3D	Yes (66)	5277	11.7	37	79	Yes

Table 1: Comparison between our dataset and existing datasets with 3D car labels. "complete 3D" means fitting with 3D car model.

severe one is that the recorded 3D LiDAR points are at best a sparse coverage of the scene from front view [21], especially for distant and absorbing regions. Since it is crucial for a self-driving car to maintain a safe breaking distance, 3D understanding from a regular camera remains a promising and viable approach attracting significant amount of research from the vision community [6, 56].

The recent tremendous success of deep convolutional network [22] in solving various computer vision tasks is built upon the availability of massive carefully annotated training datasets, such as ImageNet [11] and MSCOCO [36]. However, acquiring large-scale training datasets is an extremely laborious and expensive endeavour, and the community is especially lacking of fully annotated datasets of 3D nature. For example, for the task of 3D car understanding for autonomous driving, the availability of datasets is severely limited. Take KITTI [21] for instance. Despite being the most popular dataset for self-driving, it has only about 200 labelled 3D cars yet in the form of bounding box only, without detailed 3D shape information flow [41]. Deep learning methods are generally hungry for massive labelled training data, yet the sizes of currently available 3D car datasets are far from adequate to capture various appearance variations, *e.g.* occlusion, truncation, and lighting. For other datasets such as PASCAL3D+ [65] and ObjectNet3D [64], while they contain more images, the car instances therein are mostly isolated, imaged in a controlled lab setting, thus are unsuitable for autonomous driving.

To rectify this situation, we propose a large-scale 3D instance car dataset built from real images and videos captured in complex real-world driving scenes in multiple cities. Our new dataset, called `ApolloCar3D`, is built upon the publicly available ApolloScape dataset [23] and targets at 3D car understanding research in self-driving scenarios. Specifically, we select $5,277$ images from around 200K released images in the semantic segmentation task of ApolloScape, following several principles such as (1) containing sufficient amount of cars driving on the street, (2) exhibiting large appearance variations, (3) covering multiple driving cases at highway, local, and intersections. In addition, for each image, we provide a stereo pair for obtaining stereo disparity; and for each car, we provide 3D keypoints such as corner of doors and headlights, as well as realistic 3D CAD models with an absolute scale. An example is shown in Fig. 1(b). We will provide details about how we define those keypoints and label the dataset in Sec. 2.

Equipped with `ApolloCar3D`, we are able to directly apply supervised learning to train a 3D car understanding system from images, instead of making unnecessary compromises falling back to weak-supervision or semi-supervision like most previous works do, *e.g.* 3D-RCNN [28] or single object 3D recovery [60].

To facilitate future research based on our `ApolloCar3D` dataset, we also develop two 3D car understanding algorithms, to be used as new baselines in order to benchmark future contributed algorithms. Details of our baseline algorithms will be described in following sections.

Another important contribution of this paper is that we propose a new evaluation metric for this task, in order to to jointly measure the quality of both 3D pose estimation and shape recovery. We referred to our new metric as "Average 3D precision (A3DP)", as it is inspired by the AVP metric (average viewpoint precision) for PASCAL3D+ [65] which however only considers 3D pose. In addition, we supply multiple true positive thresholds similar to MS COCO [36].

The contributions of this paper are summarized as:

- A large-scale and growing 3D car understanding dataset for autonomous driving, *i.e.* `ApolloCar3D`, which complements existing public 3D object datasets.
- A novel evaluation metric, *i.e.* A3DP, which jointly considers both 3D shape and 3D pose thus is more appropriate for the task of 3D instance understanding.
- Two baseline algorithms for 3D car understanding, which outperform several state-of-the-art 3D object recovery methods.
- Human performance study, which points out promising future research directions.

2. ApolloCar3D Dataset

Existing datasets with 3D object instances. Previous datasets for 3D object understanding are often very limited in scale, or with partial 3D properties only, or contain few objects per image [29, 55, 52, 44, 47, 37]. For instance,

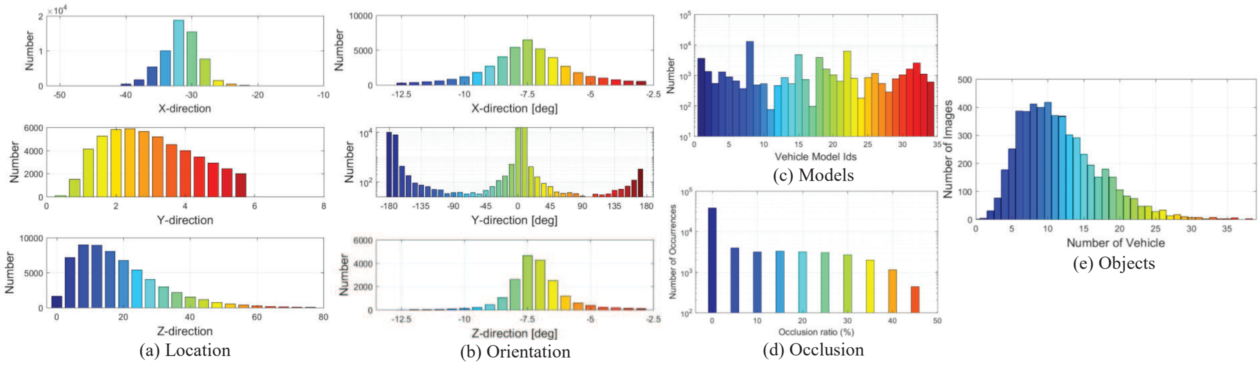

Figure 2: Car occurrence and object geometry statistics in `ApolloCar3D`. (a) and (b) illustrate the translation and orientation distribution of all the vehicles. (c) - (e) describe the distribution of vehicle type, occlusion ratio, and number of vehicles per image. Specifically, the Y-axis in all the figures represents the occurrences of vehicles.

3DObject [52] has only 10 instances of cars. The EPFL Car [47] has 20 cars under different viewpoints but was captured in a controlled turntable rather than in real scenes.

To handle more realistic cases from non-controlled scenes, datasets [35] with natural images collected from Flickr [40] or indoor scenes [10] with Kinect are extended to 3D objects [51]. The IKEA dataset [35] labelled a few hundreds indoor images with 3D furniture models. PASCAL3D+ [65] labelled the 12 rigid categories in PASCAL VOC 2012 [16] images with CAD models. ObjectNet3D [64] proposed a much larger 3D object dataset with images from ImageNet [11] with 100 categories. These datasets, while useful, are not designed for autonomous driving scenarios. To the best of our knowledge, the only real-world dataset that partially meets our requirement is the KITTI dataset [21]. Nonetheless, KITTI only labels each car by a rectangular bounding box, and lacks fine-grained semantic keypoint labels (*e.g.* window, headlight). One exception is the work of [42], yet it falls short in the number of 200 labelled images, and their car parameters are not publicly available.

In this paper, as illustrated in Fig. 1, we offer to the community the first large-scale and fully 3D shape labelled dataset with 60K+ car instances, from 5,277 real-world images, based on 34 industry-grade 3D CAD car models. Moreover, we also provide the corresponding stereo images and accurate 2D keypoint annotations. Tab. 1 gives a comparison of key properties of our dataset versus existing ones for 3D object instance understanding.

2.1. Data Acquisition

We acquire images from the ApolloScape dataset [23] due to its high resolution (3384×2710), large scale ($\geq$140K semantically labelled images), and complex driving conditions. From the dataset, we carefully select images satisfying our requirements as stated in Sec. 1. Specifically, we select images from their labelled videos of 4 different cites satisfying (1) relatively complex environment, (2) interval between selected images ≥ 10 frames. After picking images from the whole dataset using their semantic labels, in order to have more diversity, we prune all images manually, and further select ones which contain better variation of car scales, shapes, orientations, and mutual occlusion between instances, yielding 5,277 images for us to label.

For 3D car models, we look for highly accurate shape models, *i.e.* the offset between the boundary of re-projected model and manually labelled mask is less than $3px$ on average. However, 3D car meshes in ShapeNet [4] are still not accurate enough for us, and it is too costly to fit each 3D model in the presence of heavy occlusion, as shown in Fig. 1. Therefore, to ensure the quality (accuracy) of 3D models, we hired online model makers to manually build corresponding 3D models given parameters of absolute shape and scale of certain car type. Overall, we build 34 real models including sedan, coupe, minivan, SUV, and MPV, which has covered the majority of car models and types in the market.

2.2. Data Statistics

In Fig. 2, we provide statistics for the labelled cars w.r.t. translation, orientation, occlusion, and model shape. Compared with KITTI [21], `ApolloCar3D` contains significantly larger amount of cars that are at long distance, under heavy occlusions, and these cars are distributed diversely in space. From Fig. 2(b), the orientation follows a similar distribution, where the majority of cars on road are driving towards or backwards the data acquisition car. In Fig. 2(c), we show distribution w.r.t. car types, where sedans have the most frequent occurrences. The object distribution per image in Fig. 2(e) shows that most of the images contain more than 10 labeled objects.

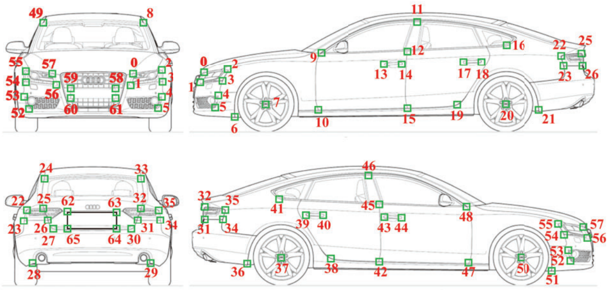

Figure 3: 3D keypoints definition for car models. 66 keypoints are defined for each model.

3. Context-aware 3D Keypoint Annotation

Thanks to the high quality 3D models that we created, we develop an efficient machine-aided semi-automatic keypoint annotation process. Specifically, we only ask human annotators to click on a set of pre-defined keypoints on the object of interest in each image. Afterwards, the EPnP algorithm [31] is employed to automatically recover the pose and model of the 3D car instance by minimizing re-projection error. RANSAC [19] is used handle outliers or wrong annotations. While only a handful of keypoints can be sufficient solve the EPnP problem, we define 66 semantic keypoints in our dataset, as shown in Fig. 3, which has much higher density than most previous car datasets [57, 43]. The redundancy enables more accurate and robust shape-and-pose registration.

Context-aware annotation. In the presence of severe occlusions, for which RANSAC also fails, we develop a context-aware annotation process by enforcing co-planar constraints between one car and its neighboring cars. By doing this, we are able to propagate information among neighboring cars, so that we jointly solve for their poses with context-aware constraints.

Formally, the objective for a single car pose estimation is

$$\mathcal{E}_{PnP}(\mathbf{p}, \mathcal{S}) = \sum_{[\mathbf{x}_k^3, k] \in \mathcal{S}} \mathbf{v}_k \|\pi(\mathbf{K}, \mathbf{p}, \mathbf{x}_k^3) - \mathbf{x}_k\|_2, \quad (1)$$

where $\mathbf{p} = [\alpha, \beta, \gamma, x, y, z] \in \mathbf{SE}(3), \mathcal{S} \in \{S_1, \cdots, S_m\}$ indicate the pose and shape of a car instance respectively. $\mathbf{v}$ is a vector indicating whether the k_{th} keypoint of the car has been labelled or not. $\mathbf{x}_k$ is the labelled 2D keypoint coordinate on the image. $\pi(\mathbf{p}, \mathbf{x}_k^3)$ is a perspective projection function projecting the correspondent 3D keypoint $\mathbf{x}_k^3$ on the car model given $\mathbf{p}$ and camera intrinsic $\mathbf{K}$.

Our context-aware co-planarity constraint is formulated as:

$$\mathcal{E}_N(\mathbf{p}, \mathcal{S}, \mathbf{p}_n, \mathcal{S}_n) = [(\alpha_\mathbf{p} - \alpha_{\mathbf{p}_n})^2 + (\beta_\mathbf{p} - \beta_{\mathbf{p}_n})^2 + ((y_\mathbf{p} - h_\mathcal{S}) - (y_{\mathbf{p}_n} - h_{\mathcal{S}_n}))^2], \quad (2)$$

where n is a spatial neighbor car, $\alpha_\mathbf{p}$ and $\beta_\mathbf{p}$ are roll and pitch component of $\mathbf{p}$, and $h_\mathcal{S}$ is the height of the car given its shape $\mathcal{S}$.

The total energy to be minimized for finding car pose and shape in image I is defined as:

$$\mathcal{E}_I = \sum_{c=1}^{C} \{\mathcal{E}_{PnP}(\mathbf{p}_c, \mathcal{S}_c) + B(\mathcal{K}_c) \sum_{n \in \mathcal{N}_c} \mathcal{E}_N(\mathbf{p}_c, \mathcal{S}_c, \mathbf{p}_n, \mathcal{S}_n)\}, \quad (3)$$

where c is the index of cars in the image, $B(\mathcal{K}_c)$ is a binary function indicating whether car c needs to borrow pose information from neighbor cars, and $\mathcal{K} = \{\mathbf{x}_k^2\}$ is the set of labelled 2D keypoints of the car. $\mathcal{N}_c = N(c, \mathbf{M}, \kappa)$ is the set of rich annotated neighboring cars of c using instance mask $\mathbf{M}$, and κ is the maximum number of neighbors we use. We list the definition details of function $B(\mathcal{K}_c)$ and $N(c, \mathbf{M}, \kappa)$ in *Supplementary Materials* due to space limit.

As illustrated in Fig. 4, to minimize Eq. (3), we first solve for those cars with dense keypoint annotations, by exhausting all car types. To guarantee the precision, we labeled the ground points of each car and use the corresponding depth map in ApolloScape dataset to get the accurate distance of each car. We require that the average re-projection error must be below 5 pixels and the obtained pose is with the minimum distance errors with the corresponding ground points. We then solve for the cars with fewer keypoint annotations, by using its context information provided by its neighboring cars, and the precision setting is the same with those cars with dense keypoint annotations. After most cars are aligned, we ask human annotators to visually verify and adjust the result before committing to the database.

4. Two Baseline Algorithms

Based on ApolloCar3D, we aim to develop strong baseline algorithms to facilitate benchmarking and future research. We first review the most recent literature and then implement two possibly strongest baseline algorithms.

Existing work on 3D instance recovery from images. 3D objects are usually recovered from multiple frames, 3D range sensors [26], or learning-based methods [67, 13]. Nevertheless, addressing 3D instance understanding from a single image in an uncontrolled environment is ill-posed and challenging, thus attracting growing attention. With the development of deep CNNs, researchers are able to achieve

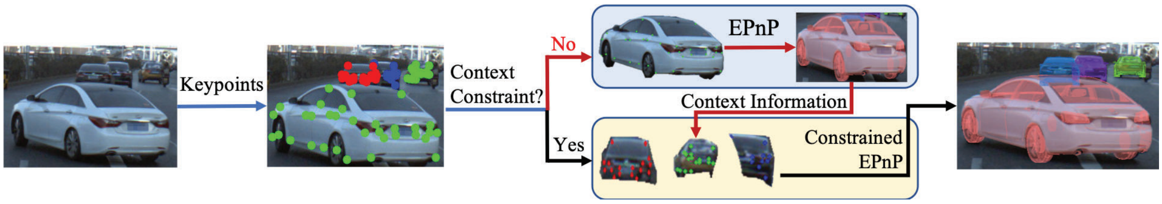

Figure 4: The pipeline for ground truth pose label generation based on annotated 2D and 3D keypoints.

impressive results with supervised [18, 69, 43, 46, 57, 54, 63, 70, 6, 32, 49, 38, 3, 66] or weakly supervised strategies [28, 48, 24]. Existing works consider to represent an object as a parameterized 3D bounding box [18, 54, 57, 49], coarse wire-frame skeletons [14, 32, 62, 69, 68], voxels [9], one-hot selection from a small set of exemplar models [3, 45, 1], and point clouds [17]. Category-specific deformable model has also been used for shapes of simple geometry [25, 24].

For handling cases of multiple instance, 3D-RCNN [28] and DeepMANTA [3] are possibly the state-of-the-art techniques by combining 3D shape model with Faster R-CNN [50] detection. However, due to the lack of high quality dataset, these methods have to rely on 2D masks or wire-frames that are coarse information for supervision. Back on ApolloCar3D, in this paper, we adapt their algorithms and conduct supervised training to obtain strong results for benchmarks. Specifically, 3D-RCNN does not consider the car keypoints, which we referred to as direct approach, while DeepMANTA considers keypoints for training and inference, which we call keypoint-based approach. Nevertheless, both algorithms are not open-sourced yet. Therefore, we have to develop our in-house implementation of their methods, serving as baselines in this paper. In addition, we also propose new ideas to improve the baselines, as illustrated in Fig. 5, which we will elaborate later.

Specifically, similar to 3D-RCNN [28], we assume predicted 2D car masks are given, *e.g.* learned through Mask-RCNN [22], and we primarily focus on 3D shape and pose recovery.

4.1. A Direct Approach

When only car pose and shape are provided, following direct supervision strategy as mentioned in 3D-RCNN [28], we crop out corresponding features for every car instance from a fully convolutional feature extractor with RoI pooling, and build independent fully connected layers to regress towards its 2D amodal center, allocentric rotation, and PCA-based shape parameters. Following the same strategy, the regression output spaces of rotation and shape are discretized. Nevertheless, for estimating depth, instead of using amodal box and enumerating depth such that the projected mask best fits the box as mentioned in [28], we use ground truth depths as supervision. Therefore, for our implementation, we replace amodal box regression to depth regression using similar depth discretizing policy as proposed in [20], which provides state-of-the-art depth estimation from a single image.

Targeting at detailed shape understanding, we further make two improvements over the original pipeline, as shown in Fig. 5(a). First, as mentioned in [28], estimating object 3D shape and pose are distortion-sensitive, and RoI pooling is equivalent to making perspective distortion of an instance in the image, which negatively impact the estimation. 3D-RCNN [28] induces infinity homography to handle the problem. In our case, we replace RoI pooling to a fully convolutional architecture, and perform per-pixel regression towards our pose and shape targets, which is simpler yet more effective. Then we aggregate all the predictions inside the given instance mask with a "self-attention" policy as commonly used for feature selection [59]. Formally, let $\mathbf{X} \in \mathbb{R}^{h\times w\times c}$ be the feature map, and the output for car instance i is computed as,

$$\mathbf{o}_i = \sum_{\mathbf{x}} \mathbf{M}^i_{\mathbf{x}}(\kappa_o * \mathbf{X} + \mathbf{b}_o)_{\mathbf{x}} \mathbf{A}_{\mathbf{x}} \tag{4}$$

where $\mathbf{o}_i$ is the logits of discretized 3D representation, $\mathbf{x}$ is a pixel in the image, $\mathbf{M}^i$ is a binary mask of object i, $\kappa_o \in \mathbb{R}^{k_l\times k\times c\times b}$ is the kernels used for predicting outputs, and $\mathbf{A} \in \mathbb{R}^{h\times w\times 1}$ is the attention map. b is the number of bins for discretization following [28]. We call feature aggregation as mask pooling since it selects the most important information within each object mask.

Secondly, as shown in our pipeline, for estimating car translation, *i.e.* its amodal center $\mathbf{c}_a = [c_x, c_y]$ and depth d_c, instead of using the same target for every pixel in a car mask, we propose to output a 3D offset at each pixel w.r.t. the 3D car center, which provides stronger supervision and helps learn more robust networks. Previously, inducing relative position of object instances has also been shown to be effective in instance segmentation [58, 33]. Formally, let $\mathbf{c} = [d_c(c_x - u_x)/f_x, d_c(c_y - u_y)/f_y, d_c]$ be the 3D car center, and our 3D offset for a pixel $\mathbf{x} = [x, y]$ is defined as $\mathbf{f}^3 = \mathbf{x}^3 - \mathbf{c}$, where $\mathbf{x}^3 = [d(x - u_x)/f_x, d(y - u_y)/f_y, d]$, and d is the estimated depth at $\mathbf{x}$. In principle, 3D offset estimation is equivalent to jointly computing per-pixel 2D offset respect to the amodal center, *i.e.* $\mathbf{x} - \mathbf{c}_a = [u, v]^T$ and a relative depth to the center depth, *i.e.* $d - d_c$. We

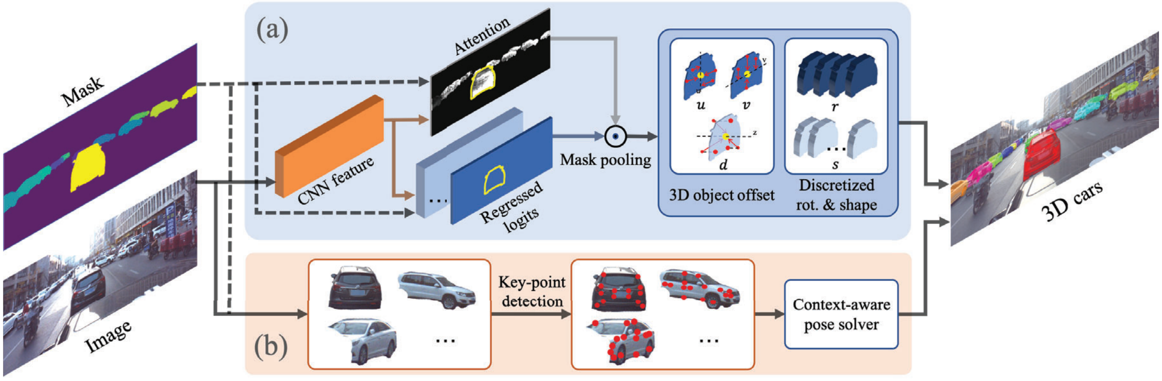

Figure 5: Training pipeline for 3D car understanding. Upper (a): direct approach. Bottom (b): key point based approach.

adopt such a factorized representation for model center estimation, and the 3D model center can then be recovered by

$$\mathbf{c}_a = \sum_{\mathbf{x}} \mathbf{A}_{\mathbf{x}}(\mathbf{x} + \mathbf{f}^3_{x,y}), d_c = \sum_{\mathbf{x}} \mathbf{A}_{\mathbf{x}}(d_{\mathbf{x}} + \mathbf{f}^3_d) \quad (5)$$

where $veA_{\mathbf{x}}$ is the attention at $\mathbf{x}$, which is used for output aggregation in Eq. (4). In our experiments in Sec. 5, we show that the two strategies provide improvements over the original baseline results.

4.2. A Keypoint-based Approach

When sufficient 2D keypoints from each car are available (*e.g.*as in Fig. 5(b)), we develop a simple baseline algorithm, inspired by DeepMANTA [3], to align 3D car pose via 2D-3D matching.

Different from [3], our 3D car models have much more geometric details and come with the absolute scale, and our 2d keypoints have more precise annotations. Here, we adopt the CPM [61] – a state-of-the-art 2d keypoint detector despite the algorithm was originally developed for human pose estimation. We extend it to 2d car keypoint detection and find it works well.

One advantage of using 2d keypoint prediction over our baseline-1 *i.e.*the "direct approach" in Sec. 4.1, is that, we do not have to regress the global depth or scale – the estimation of which by networks is in general not very reliable. Instead of feeding the full image into the network, we crop out each car region in the image for 2d keypoint detection. This is especially useful for images in ApolloScape [23], which have a large number of cars of small size.

Borrowing the context-aware constraints from our annotation process, once we have enough detected keypoints, we first solve the easy cases where a car is less occluded using EPnP[31], then we propagate the information to neighboring cars until all car pose and shapes are found to be consistent with each other w.r.t. the co-planar constraints via optimizing Eq. (3). We referred our car pose solver with co-planar constraints as **context-aware solver**.

5. Experiments

This section provides implementation details, our newly proposed evaluation metric, and experiment results. In total, we have experimented on 5,277 images, split to 4,036 for training, 200 for validation, and 1,041 for testing.

Implementation details. Due to the lacking of publicly available source codes, we re-implemented 3D-RCNN [28] for 3D car understanding without using keypoints, and DeepMANTA [3] which requires key points annotation. For training Mask-RCNN, we downloaded the code from GitHub implemented by an autonomous driving company [1]. We adopted the fully convolutional features from DeepLabv3 [5] with Xception65 [8] network and follow the same training policy. For DeepMANTA, we used the key point prediction methods from CPM [7]. With 4,036 training images, we obtained about 40,000 labeled vehicles with 2D keypoints, used to train a CPM [7] (with 5 stages of CPM, and VGG-16 initialization).

Evaluation metrics. The average precision (AP) [16] is usually used for evaluating 3D object understanding. And, the similarity is measured using 3D bounding box IoU [21] with orientation (average orientation similarity (AOS) [21]) or 2D bounding box with viewpoint (average viewpoint precision (AVP) [65]). Those metrics only measure coarse 3D properties, without considering the influence of object shape.

Mesh distance [53] and voxel IoU [12] are usually used to evaluate 3D shape reconstruction. In our case, a car model is mostly compact, thus we consider comparing projection masks of two models following the idea of visual hull representation [39]. Specifically, we sample 100 orientations at yaw angular direction and project each view of the model to an image with a resolution of 1280×1280. We use the mean IoU over all views as the car shape similarity metric. For evaluating rotation and translation, we follow the metrics commonly used for camera pose estimation [21]. In

[1] https://github.com/TuSimple/mx-maskrcnn

Methods	Mask	wKP	A3DP-Abs			A3DP-Rel			Time(s)
			mean	c-l	c-s	mean	c-l	c-s	
3D-RCNN* [28]	gt	-	16.44	29.70	19.80	10.79	17.82	11.88	0.29s
+ MP	gt	-	16.73	29.70	18.81	10.10	18.81	11.88	0.32s
+ MP + OF	gt	-	17.52	30.69	20.79	13.66	19.80	13.86	0.34s
+ MP + OF	pred.	-	15.15	28.71	17.82	11.49	17.82	11.88	0.34s
DeepMANTA* [3]	gt	✓	20.10	30.69	23.76	16.04	23.76	19.80	3.38s
+ CA-solver	gt	✓	**21.57**	**32.62**	**26.73**	**17.52**	**26.73**	**20.79**	7.41s
+ CA-solver	pred.	✓	20.40	31.68	24.75	16.53	24.75	19.80	8.5s
Human	gt	✓	38.22	56.44	49.50	33.27	51.49	41.58	607.41s

Table 2: Comparison among baseline algorithms. $*$ means in-house implementation. "Mask" means the provided mask for 3D understanding ("gt" means ground truth mask and "pred." means Mask-RCNN mask). "wKP" means using keypoint predictions. "c-l" indicates results from loose criterion, and "c-s" indicates results from strict criterion. "MP" stands for mask pooling and "OF" stands for offset flow. "CA-solver" stands for context-aware 3D pose solver. "Times(s)" indicates the average inference times cost for processing each image.

summary, the criteria for judging a true positive given a set of thresholds is defined as

$$\begin{aligned} c_{shape} &= \frac{1}{|V|}\sum\nolimits_{v\in\mathcal{V}} IoU(\mathbf{P}(s_i), \mathbf{P}(s_i^*))_v \geq \delta_s, \\ c_{trans} &= |\mathbf{t}_i - \mathbf{t}_i^*|_2 \leq \delta_t, \\ c_{rot} &= \arccos(|\mathbf{q}(\mathbf{r}_i)\cdot\mathbf{q}(\mathbf{r}_i^*)|) \leq \delta_r, \end{aligned} \tag{6}$$

where $s, \mathbf{t}, \mathbf{r}$ are the shape ID, translation, and rotation of a predicted 3D car instance.

In addition, a single set of true positive thresholds used by AOS or AVP, *e.g.* IoU ≥ 0.5, and rotation $\leq \pi/6$, is not sufficient to evaluate detected results thoroughly [21]. Here, following the metric of MS COCO [36], we propose to use multiple sets of thresholds from loose to strict for evaluation. Specifically, the thresholds used in our results for all levels of difficulty are $\{\delta_s\} = [0.5 : 0.05 : 0.95], \{\delta_t\} = [2.8 : 0.3 : 0.1], \{\delta_r\} = [\pi/6 : \pi/60 : \pi/60]$, where $[a : i : b]$ indicates a set of discrete thresholds sampled in a line space from a to b with an interval of i. Similar to MSCOCO, we select one loose criterion $\mathbf{c}-l = [0.5, 2.8, \pi/6]$ and one strict criterion $\mathbf{c}-l = [0.75, 1.4, \pi/12]$ to diagnose the performance of different algorithms. Note that in our metrics, we only evaluate instances with depth less than $100m$ as we would like to focus on cars that are more immediately relevant to our autonomous driving task.

Finally, in self-driving scenarios that are safety critical, we commonly care nearby cars rather than those far away. Therefore, we further propose to use a relative error metric for evaluating translation following the "AbsRel" commonly used in depth evaluation [21]. Formally, we change the criteria of c_{trans} to $|\mathbf{t}_i - \mathbf{t}_i^*|/\mathbf{t}_i^* \leq \delta_t^*$, and set the thresholds to $\{\delta_t^*\} = [0.10 : 0.01 : 0.01]$. We call our evaluation metric with absolute translation thresholds as "A3DP-Abs", and the one with relative translation thresholds as "A3DP-Rel".

5.1. Quantitative Results

In this section, we compare against our baseline algorithms with the method presented in Sec. 4 by progressively adding our proposed components and losses. Tab. 2 shows the comparison results. For direct regression approach, our baseline algorithm "3D-RCNN" provides regression towards translation, allocentric rotation, and car shape parameters. We further extend the baseline method by adding mask pooling (MP) and offset flow (OF). We observe from the table that, swapping RoI pooling for mask pooling moderately improves the results while offset flow brings significant boost. They together help avoiding geometric distortions from regular RoI pooling and bring attention mechanism to focus on relevant regions.

For the keypoint-based method, "DeepMANTA" shows the results by using our detected key points and solving with PnP for each car individually, yielding reasonable performance. "+CA-solver" means for cars without sufficient detected key points, we employ our context-aware solver for inference, which provides around 1.5% improvement. For both methods, switching ground truth mask to segmentation from Mask R-CNN gives little drop of the performance, demonstrating the high quality of Mask R-CNN results.

Finally, we train a new group of labellers, and ask them to re-label the keypoints on our validation set, which are passed through our context-aware 3D solver. We denote these results as "human" performance. We can see there is a clear gap ($\sim 10\%$) between algorithms with human. However, even the accuracy for humans is still not satisfying. After checking the results, we found that this is primarily because humans cannot accurately memorize the semantic meaning of all the 66 keypoints, yielding wrongly solved poses. We conjecture this could be fixed by rechecking and refinement, possibly leading to improved performance.

5.2. Qualitative Results

Some qualitative results are visualized in Fig. 7. From the two examples, we can find that the additional key point predictions provide more accurate 3D estimation than direct method due to the use of geometric constraints and inter-car relationship constraints. In particular, for the direct method, most errors occur in depth prediction. It can be explained by the nature of the method that the method predicts the global 3D property of depth purely based on object

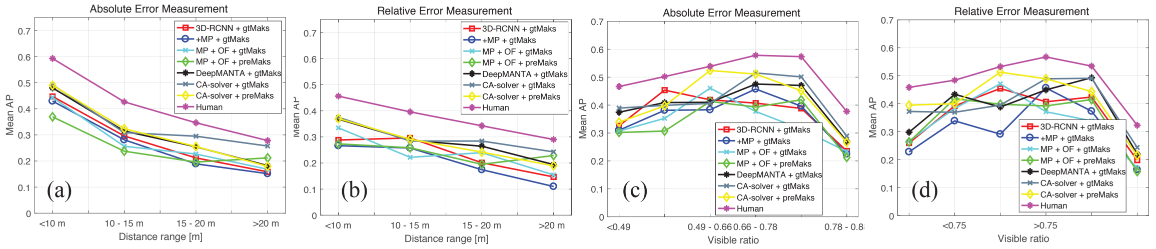

Figure 6: 3D understanding results of various algorithms w.r.t. different factors causing false estimation. (a) A3DP-Abs v.s distance, (b) A3DP-Rel v.s distance, (c) A3DP-Abs v.s occlusion, (d) A3DP-Abs v.s occlusion.

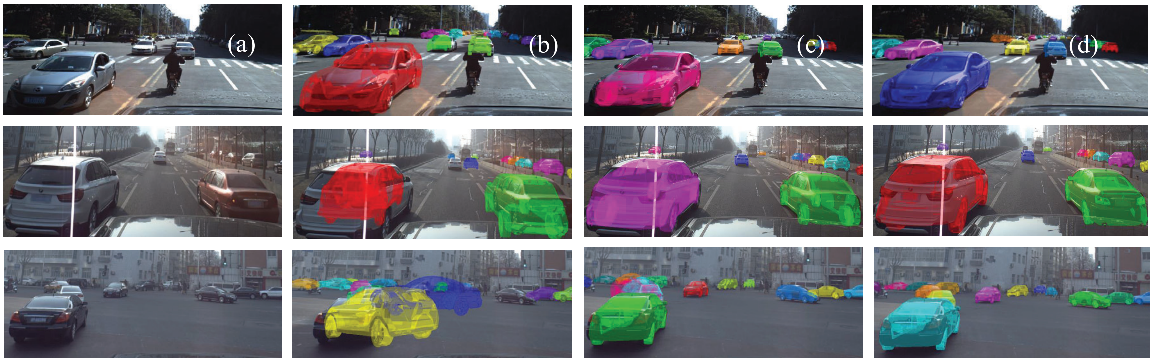

Figure 7: Visualization results of different approaches, in which (a) the input image, (b) and (c) are the results with direct regression method and key points-based method with context constraint. (d) gives the ground truth results.

appearance in 2D, which is ill-posed and error-prone. However, thanks to the use of reliable masks, the method discovers more cars than the keypoint-based counterpart. For the keypoint-based approach, we are able to show that correctly detected keypoints are extremely successful at constraining car poses, while failed or missing keypoint estimation, especially for cars of unusual appearance, will lead to missing detection of cars or wrong solution for poses.

5.3. Result Analysis

To analyze the performance of different approaches, we evaluate them separately on various distances and occlusion ratios. Detailed results are shown in Fig. 6. Checking Fig. 6(a, b), as expected, we can find that the estimation accuracy decreases with farther distances, and the gap between human and algorithm narrows in the distance. In addition, after checking Fig. 6(c, d) for occlusion, we discover that the performance also drops with increasing the occlusion ratio. However, we observe that the performance on non-occluded cars is the worst on average among all occlusion patterns. This is because most cars which experience little occlusion are from large distance and of small scale, while cars close-by are more often occluded.

6. Conclusion

This paper presents by far the largest and growing dataset (namely `ApolloCar3D`) for instance-level 3D car understanding in the context of autonomous driving. It is built upon industrial-grade high-precision 3D car models fitted to car instances captured in real world scenarios. Complementing existing related datasets *e.g.* [21], we hope this new dataset could serve as a long-standing benchmark facilitating future research on 3D pose and shape recovery.

In order to efficiently annotate complete 3D object properties, we have developed a context-aware 3D annotation pipeline, as well as two baseline algorithms for evaluation. We have also conducted carefully designed human performance study, which reveals that there is still a visible gap between machine performance and that of human's, motivating and suggesting promising future directions. More importantly, built upon the publicly available ApolloScape dataset [23], our `ApolloCar3D` dataset contains multitude of data sources including stereo, camera pose, semantic instance label, per-pixel depth ground truth, and moving videos. Working with our data enables training and evaluation of a wide range of other vision tasks, *e.g.* stereo vision, model-free depth estimation, and optical flow *etc.*, under real scenes.

7. Acknowledgement

The authors gratefully acknowledge He Jiang from Baidu Research for his help in car visualization, Xinyu Huang from Baidu Research for his help in keypoints definition, Shichao Zhao from Baidu for his help in 2d keypoints detection.

References

[1] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014.

[2] G. Canal, S. Escalera, and C. Angulo. A real-time human-robot interaction system based on gestures for assistive scenarios. *Computer Vision and Image Understanding*, 149:65–77, 2016.

[3] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2040–2049, 2017.

[4] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

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

[7] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, and G. Y. J. Sun. Cascaded pyramid network for multi-person pose estimation.

[8] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, pages 1610–02357, 2017.

[9] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. Eur. Conf. Comp. Vis.*, 2016.

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 248–255. Ieee, 2009.

[12] X. Di and P. Yu. 3d reconstruction of simple objects from a single view silhouette image. *arXiv preprint arXiv:1701.04752*, 2017.

[13] N. Dinesh Reddy, M. Vo, and S. G. Narasimhan. Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2018.

[14] W. Ding, S. Li, G. Zhang, X. Lei, H. Qian, and Y. Xu. Vehicle pose and shape estimation through multiple monocular vision.

[15] F. Engelmann, J. Stückler, and B. Leibe. Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors. In *German Conference on Pattern Recognition*, pages 219–230. Springer, 2016.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[17] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

[18] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 611–619, 2012.

[19] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[20] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[21] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3354–3361. IEEE, 2012.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2980–2988. IEEE, 2017.

[23] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 954–960, 2018.

[24] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *Proc. Eur. Conf. Comp. Vis.*, 2018.

[25] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1966–1974, 2015.

[26] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *Proc. Eur. Conf. Comp. Vis.*, pages 205–220. Springer, 2016.

[27] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 2539–2547, 2015.

[28] A. Kundu, Y. Li, and J. M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3559–3568, 2018.

[29] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, volume 2, pages II–409, 2003.

[30] J. J. Leonard and H. F. Durrant-Whyte. *Directed sonar sensing for mobile robot navigation*, volume 175. Springer Science & Business Media, 2012.

[31] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *Int. J. Comp. Vis.*, 81(2):155, 2009.

[32] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. *arXiv preprint arXiv:1612.02699*, 2016.

[33] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. *arXiv preprint arXiv:1611.07709*, 2016.

[34] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proc. Eur. Conf. Comp. Vis.*, pages 641–656, 2018.

[35] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2992–2999, 2013.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, pages 740–755. Springer, 2014.

[37] R. Lopez-Sastre, C. Redondo-Cabrera, P. Gil-Jimenez, and S. Maldonado-Bascon. Icaro: image collection of annotated real-world objects, 2010.

[38] F. Massa, R. Marlet, and M. Aubry. Crafting a multi-task cnn for viewpoint estimation. *arXiv preprint arXiv:1609.03894*, 2016.

[39] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374. ACM Press/Addison-Wesley Publishing Co., 2000.

[40] J. McAuley and J. Leskovec. Image labeling on a network: using social-network metadata for image classification. In *Proc. Eur. Conf. Comp. Vis.*, pages 828–841. Springer, 2012.

[41] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3061–3070, 2015.

[42] M. Menze, C. Heipke, and A. Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[43] Y. Miao, X. Tao, and J. Lu. Robust 3d car shape estimation from landmarks in monocular image. In *Proc. Brit. Mach. Vis. Conf.*, 2016.

[44] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *Int. J. Comp. Vis.*, 73(3):263–284, 2007.

[45] R. Mottaghi, Y. Xiang, and S. Savarese. A coarse-to-fine model for 3d pose estimation and sub-category recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 418–426, 2015.

[46] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3d bounding box estimation using deep learning and geometry. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5632–5640. IEEE, 2017.

[47] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 778–785. IEEE, 2009.

[48] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2018.

[49] P. Poirson, P. Ammirato, C.-Y. Fu, W. Liu, J. Kosecka, and A. C. Berg. Fast single shot detection and pose estimation. In *3dv*, pages 676–684. IEEE, 2016.

[50] 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*, pages 91–99, 2015.

[51] B. C. Russell and A. Torralba. Building a database of 3d scenes from user annotations. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2711–2718. IEEE, 2009.

[52] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, Oct 2007.

[53] D. Stutz and A. Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018.

[54] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2686–2694, 2015.

[55] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. V. Gool. Towards multi-view object class detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, volume 2, pages 1589–1596, June 2006.

[56] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 4489–4497, 2015.

[57] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1510–1519, 2015.

[58] J. Uhrig, E. Rehder, B. Fröhlich, U. Franke, and T. Brox. Box2pix: Single-shot instance segmentation by assigning pixels to object boxes. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[60] Y. Wang, X. Tan, Y. Yang, X. Liu, E. Ding, F. Zhou, and L. S. Davis. 3d pose estimation for fine-grained object categories. *arXiv preprint arXiv:1806.04314*, 2018.

[61] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[62] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3d interpreter network. In *ECCV*, pages 365–382. Springer, 2016.

[63] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Data-driven 3d voxel patterns for object category recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1903–1911, 2015.

[64] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese. Objectnet3d: A large scale database for 3d object recognition. In *Proc. Eur. Conf. Comp. Vis.*, pages 160–176. Springer, 2016.

[65] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. pages 75–82. IEEE, 2014.

[66] G. Yang, Y. Cui, S. Belongie, and B. Hariharan. Learning single-view 3d reconstruction with limited pose supervision. In *Proc. Eur. Conf. Comp. Vis.*, September 2018.

[67] T. Yu, J. Meng, and J. Yuan. Multi-view harmonized bilinear network for 3d object recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2018.

[68] M. Zeeshan Zia, M. Stark, and K. Schindler. Are cars just 3d boxes?-jointly estimating the 3d shape of multiple objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3678–3685, 2014.

[69] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3d representations for object recognition and modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2608–2623, 2013.

[70] M. Z. Zia, M. Stark, and K. Schindler. Towards scene understanding with detailed 3d object representations. *Int. J. Comp. Vis.*, 112(2):188–203, 2015.