Robust 2D/3D Vehicle Parsing in Arbitrary Camera Views for CVIS

Hui Miao[† 1], Feixiang Lu[† 2], Zongdai Liu[2], Liangjun Zhang[2], Dinesh Manocha[3] and Bin Zhou[* 1,4]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[2]Robotics and Autonomous Driving Laboratory, Baidu Research
[3]University of Maryland, College Park   [4]Peng Cheng Laboratory, Shenzhen, China

Abstract

We present a novel approach to robustly detect and perceive vehicles in different camera views as part of a cooperative vehicle-infrastructure system (CVIS). Our formulation is designed for arbitrary camera views and makes no assumptions about intrinsic or extrinsic parameters. First, to deal with multi-view data scarcity, we propose a part-assisted novel view synthesis algorithm for data augmentation. We train a part-based texture inpainting network in a self-supervised manner. Then we render the textured model into the background image with the target 6-DoF pose. Second, to handle various camera parameters, we present a new method that produces dense mappings between image pixels and 3D points to perform robust 2D/3D vehicle parsing. Third, we build the first CVIS dataset for benchmarking, which annotates more than 1540 images (14017 instances) from real-world traffic scenarios. We combine these novel algorithms and datasets to develop a robust approach for 2D/3D vehicle parsing for CVIS. In practice, our approach outperforms SOTA methods on 2D detection, instance segmentation, and 6-DoF pose estimation by 3.8%, 4.3%, and 2.9%, respectively.

1. Introduction

Cooperative vehicle-infrastructure system (CVIS) have become a key focus of research and technology in the field of autonomous driving (AD) [43, 41]. In a CVIS, camera, radar, LiDAR, and other sensors are mounted on vehicles and street-light poles at different locations (*e.g.*, front, side, top, *etc.*). The simultaneous perception of vehicles and road terminals can minimize blind zones and provide warning for out-of-sight collisions in advance. An example is shown in Fig. 1, which is a typical traffic scenario. From the front-view of the autonomous vehicle, many of the other vehi-
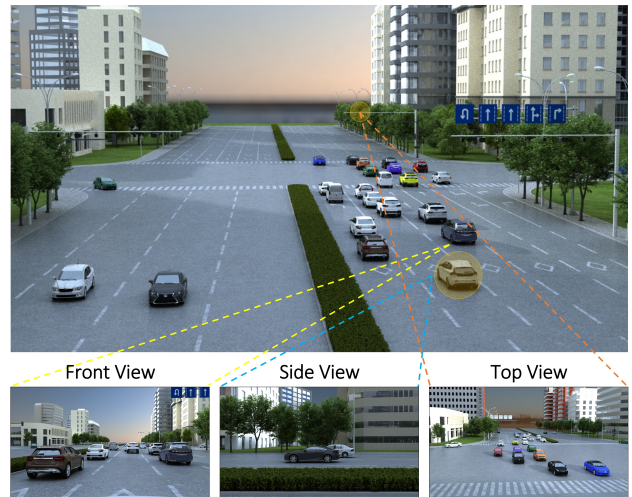

Figure 1: We highlight a scene rendered from multiple views in a CVIS. We use our approach to detect and parse novel-view vehicles from a single image corresponding to the front view.

cles are partially or completely occluded, presenting perception challenges. From the side/top-view, we can clearly see many of these occluded vehicles' views. Although multiple views can get better perception results, in the end, performance relies on improvement *w.r.t.* each of these single views. The ultimate perception problem in a CVIS, therefore, is *how to effectively detect the vehicles in novel views*. Many widely used AD datasets (*e.g.*, KITTI [15], CityScapers [7], ApolloScape [18], ApolloCar3D [45]) only provide labeled front-view data. If we directly use these labeled front-view datasets to train the deep neural networks (DNNs), the detection performance would degrade dramatically when testing these networks on the data from other views (*e.g.*, the side view or the top-view). A common strategy is to capture new images for manual annotation, which is labor-intensive, costly, and inefficient. As a result, we need a novel set of view synthesis algorithms for data

augmentation that can enable us to perform robust 2D/3D vehicle parsing for CVIS.

To address the data scarcity challenge, we propose augmenting existing AD datasets via novel view synthesis. In computer vision, view synthesis has been extensively studied. Researchers use 3D model rendering techniques (*e.g.*, [46], [39]), image-based appearance flow approaches (*e.g.*, [63], [64]), or generative adversarial network (GAN) (*e.g.*, [65], [44]) to synthesize novel-view images. Although they can achieve good results, these methods have several limitations in the context of CVIS. First, their synthesized results are mostly used for visualization, which are difficult for deep networks to learn due to the domain gap. Second, it is difficult to obtain ground-truth annotations, especially 3D information (*e.g.*, a 3D bounding box and a 6-DoF pose). Third, these approaches rely on multi-view or paired images as guidance for training, which can be obtained by 3D model rendering or a camera array in the laboratory setting. However, it is difficult to obtain such data in real AD scenarios due to occlusions and fast vehicle motion.

Some of these synthesis problems can also be addressed using techniques for 3D parsing from single images. 3D parsing from a single image is important for AD, but remains challenging because pinhole cameras cannot obtain absolute 3D positions due to projective mapping. Many state-of-the-art (SOTA) methods (*e.g.*, [30], [3], [9]) require training a depth estimation network using the ground-truth depth map or stereo pairs, with fixed intrinsic and extrinsic parameters for the cameras. They can get good results on the training dataset, but their generalizability and robustness are limited because 1) it is difficult to obtain the depth map in the real world, especially the depth of the background and 2) cameras have distinct intrinsic and extrinsic parameters. Therefore, it is difficult for the trained model in camera $\mathcal{A}$ to estimate the 6-DoF vehicle pose in camera $\mathcal{B}$.

1.1. Main Contributions

In this paper, we address the problem of data augmentation to achieve novel-view 2D/3D vehicles parsing for CVIS. We address two main challenges: 1) automatic view synthesis for data augmentation and ground-truth 2D/3D labels generation in novel views and 2) robust 3D parsing from a single image in novel views. We present a novel approach to detect and parse novel-view vehicles from a single image. This includes a new method for data augmentation of front-views in AD datasets and a novel robust approach for 2D/3D vehicle parsing. The key innovation of our data augmentation approach is the use of part-based texture inpainting for novel view synthesis. Specifically, we use the existing vehicle-based datasets that provide 3D vehicle templates and associated 6-DOF datasets. We first project the image pixels to the texture map of the template. Because there could be some holes or blank regions due to the pro-

jective mapping, we then train a part-based texture inpainting network to fill the missing data. After obtaining the complete texture map, we render the textured 3D templates with arbitrary 6-DoF poses into novel views.

Based on this synthesized data generated from vehicle templates, we present a new approach for 2D/3D parsing that is robust for arbitrary camera parameters. Instead of directly learning depth from images, the key idea of our approach is learning dense mappings between the image pixels and the canonical 3D vehicle template. Specifically, we design a multi-task network that outputs results of 2D detection, instance segmentation, a 3D bounding box (*i.e.* width, height, and length), and canonical 3D points. These canonical 3D points are one-to-one mapped to the image pixels; thus, we can compute the 6-DoF pose by solving the RANSAC-PnP problem using the input intrinsic and extrinsic parameters of the camera.

Finally, to benchmark our synthesized data and 2D/3D parsing approach, we have constructed, to the best of our knowledge, the first CVIS-oriented dataset with vehicles in different camera views, which contains 1540 labeled images and 14017 2D vehicle instances. For each vehicle instance in our dataset, we annotate its 2D bounding box, instance-level segmentation, 3D bounding box, and 6-DoF pose.

In summary, our contributions include:

1) We present a novel and compact data augmentation pipeline for vehicle parsing in CVIS. This includes synthesized data generation, data learning for 2D/3D parsing, and real-world dataset construction for benchmarking.

2) A 3D-assisted image augmentation approach is proposed to handle novel views. The key innovation is a part-based texture inpainting network with a self-supervised approach, which can automatically synthesize novel-view images with ground-truth annotations.

3) For robust 2D/3D vehicle parsing, we learn dense correspondences between the image pixels and the 3D points in a 3D vehicle template, which is robust for 6-DoF pose estimation under arbitrary camera parameters.

4) We build a real-world CVIS dataset with 2D/3D vehicle annotation for benchmarking. We compare with other SOTA methods of data augmentation and 2D/3D vehicle parsing. We highlight the accuracy improvements.

2. Related Work

2.1. Data Augmentation

The "fuel" of deep networks is labeled datasets. The most common strategy for generating enough data to train a model is manually crowd sourcing real images to annotate, which is labor intensive [16]. To decrease overhead and improve efficiency, researchers have recently focused on data augmentation techniques. Existing approaches can be categorized into four classes: 1) *3D model-based approaches*;

2) *image editing approaches*; 3) *appearance flow-based approaches*; and 4) *GAN-based approaches*.

With the publication of large-scale 3D model datasets [5, 8, 32], we can render these 3D models to generate images with ground-truth annotations such as view-point [46], segmentation [39], and depth map [42]. Based on the rendering data, explicit methods [22, 6, 21, 62, 39, 52, 20] and implicit methods [55, 50, 54, 59] are developed to synthesize the novel-view images. However, rendering is a time-consuming process that requires a lot of human interactions such as pre-building 3D scenes [19, 11]. In addition, it is not easy to reduce the domain gap between the rendering data (CG style) and the real captured data (photo), though there are some domain adaptation methods (*e.g.*, [49]).

Another approach for data augmentation is image editing [26]. Dwibedi *et al.* [10] cut objects from images and then paste them to other backgrounds to synthesize photo-realistic training data. However, these "cut-paste" data cannot rotate or translate in the 3D space. Moreover, this method is limited to handling occlusion problems.

Appearance flow-based approaches [63, 35, 64] use dense pixel-to-pixel correspondences to generate the novel-view images directly, which can be regarded as optical flow prediction. However, these methods are supervised on the target images, requiring controlled environments (*e.g.*, camera pose, scene geometry). For AD, it is difficult to obtain such dense correspondences from street-view images.

Recently, GAN-based approaches [65, 51, 2, 57, 53, 29] have been proposed to generate photo-realistic images using a generator-discriminator architecture. However, GAN-based methods have some limitations. First, GAN results are not controllable or robust. Second, as with the majority of deep networks, their results largely depend on the training data. More importantly, it is difficult to generate ground-truth 3D annotations (*e.g.*, 6-DoF pose) for training.

2.2. Image-based Vehicle Parsing in AD

For AD, it is important to parse the vehicles as 2D and 3D objects. Prior techniques can be divided into three categories: 1) *2D detection* (*e.g.*, SSD513 [14], YOLOv3 [38], Faster-RCNN [40]); 2) *instance segmentation* (*e.g.*, Mask-RCNN [17]); and 3) *6-DoF pose estimation or 3D detection* (*e.g.*, AM3D [30], DPOD [60], PV-Net [36], D4LCN [9]). These prior works can get good results in the current camera view-point, but their performance degrades in different or arbitrary views. In other words, it is difficult for their trained models in camera $\mathcal{A}$ to parse vehicles in camera $\mathcal{B}$.

2.3. Vehicle Re-Identification

Our approach is related to the prior works on vehicle re-identification (Re-ID) [56, 25, 24, 31], which aims to identify all the images of vehicles with the same vehicle ID in different camera views. However, the Re-ID images

are cropped for classification/retrieval/categorization and do not have 2D instance or 6-DoF pose annotations. Thus, existing vehicle Re-ID methods are limited to instance-level segmentation and 6-DoF pose estimation, though Re-ID is an essential module in the context of CVIS.

2.4. Datasets for CVIS

Recently, several datasets have been constructed and released for 2D/3D perception in autonomous driving (*e.g.*, KITTI [15], CityScapes [7], Mapillary [34], ApolloScape [18], and ApolloCar3D [45]). However, these data are captured from the front view, meaning the trained model is limited to the test data on the side view or the top view. For CVIS, we focus more on 2D/3D vehicle perception from multiple camera views. The AI-City-Challenge [33] provides city-scale multi-camera images/videos for vehicle Re-ID [47], tracking [48], and retrieval [13], which are important to make transportation systems "smart" and efficient. However, these data and tasks focus more on 2D vehicle parsing [58], while our goal is 2D/3D parsing, particularly 6-DoF poses. We believe 3D information is critical to CVIS, enabling safer roads and reducing traffic jams.

3. Method

In this section, we present a novel and compact data augmentation pipeline for 2D/3D vehicle parsing, which includes three main components:

1) *Data* (Sec. 3.1). We present a part-assisted data augmentation method to synthesize images and generate ground-truth annotations for the training network. The key innovation is the part-based inpainting network (Sec. 3.1.1).

2) *Learning* (Sec. 3.2). Based on our synthesized data, we present a robust approach to perform 2D/3D parsing.

3) *Evaluation* (Sec. 3.3). We construct a real-world, novel dataset for benchmarking on 2D/3D tasks.

3.1. Part-Assisted Data Augmentation

To generate the novel-view data for the training network, we propose a novel part-assisted data augmentation approach (Fig. 2). The inputs are the 3D vehicle template and associated 6-DoF pose datasets (*e.g.*, [45], [28]) and the background images with camera calibration. The output is the synthesized novel-view images with ground-truth 2D/3D annotations. In a word, our approach can be divided into three main modules: *1) foreground generation*, *2) background processing*, and *3) synthesized images with 2D/3D ground-truth annotations*.

In this paper, for the foreground, we focus on vehicle. The input is the labeled 2D-3D vehicle dataset, which annotates the 6-DoF pose and the real-size 3D shape (deformed from the parametric geometric vehicle representation) for each 2D vehicle instance. As shown in the top-right of
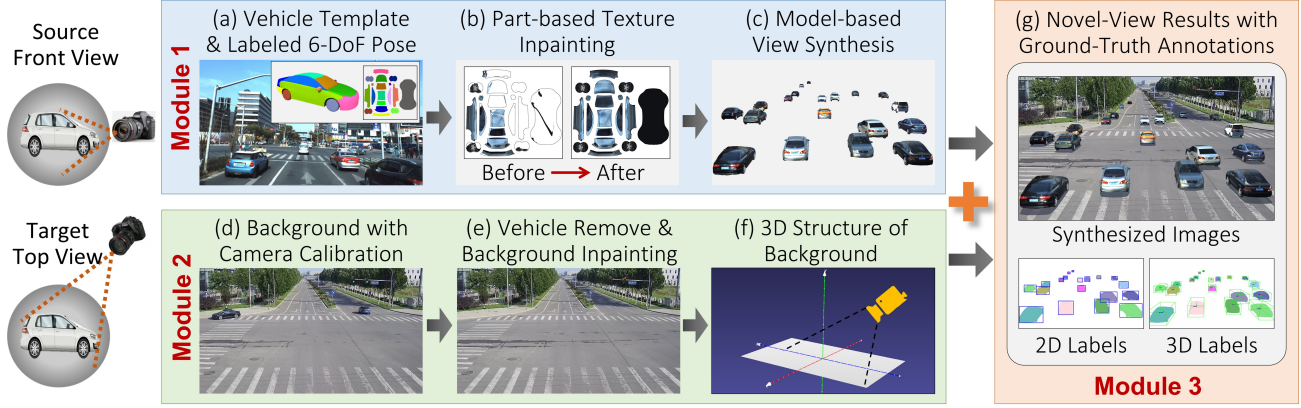
Figure 2: Our data augmentation approach includes three main modules. 1) *Foreground*: (a) We use a part-level vehicle template representation to synthesize the foreground (*i.e.* vehicles). Based on the labeled 6-DoF pose between the 2D instance and the 3D template, we can project the image pixels to the texture map with missing regions. (b) A part-based texture inpainting network is proposed to synthesize a complete texture map. (c) Then we can render the textured vehicle template to generate novel-view images. 2) *Background*: (d) We use the real traffic images with the pre-calibrated intrinsic and extrinsic parameters. (e) The moving objects are inpainted to generate a clean background. (f) We further compute the 3D structure of the background using the camera parameters. 3) *Annotations*: (g) We render the textured vehicle template to the background to synthesize color images with ground-truth 2D/3D annotations.

Fig. 2 (a), the parametric geometric vehicle representation has two features. First, it is a *geometric model* with an unfolded texture map and multiple uniform vehicle parts (*e.g.*, door, body, trunk). Second, it is also a *parametric model* with the PCA representation, where any new vehicle model can be represented as a linear combination of several principal components with coefficients. Based on the labeled 6-DoF pose and the camera parameters, we can project the 3D template to the image plane. Thus, the image pixels can be mapped to the texture map of the 3D template, which inevitably yields a lot of missing regions due to projective mapping (Fig. 2 (b)). Then we design a part-based texture inpainting network to fill these missing regions (details in Sec. 3.1.1). After that, we can obtain complete texture maps for 3D model rendering with different intrinsic and extrinsic parameters of cameras under various illuminations (Fig. 2 (c)). Note that our vehicle data augmentation approach is generic and can be used to handle pedestrians. Such objects can be constructed from parts, and we can use part-based PCA templates (*e.g.*, *SMPL* [27]).

For the background, we capture a lot of images from multiple views, especially the side-view and the top-view, where the camera's intrinsic and extrinsic parameters are pre-calibrated (Fig. 2 (d)). Next, we detect the existing vehicles and then remove them through the inpainting method (*i.e.* [61]) to generate a clean background (Fig. 2 (e)). Then we estimate the 3D structure of the background (*i.e.* the normal of the road plane) based on the camera's intrinsic and extrinsic parameters (Fig. 2 (f)).

Based on the "clean" background image and the associ-

ated 3D structure (module 2), we randomly put the 3D textured vehicles (module 1) with collision avoidance through intersection detection of 3D bounding boxes. Then we render the posed vehicle model to the background images. To enhance the fidelity of the synthesized results, we further generate the vehicle shadows according to the environment illumination and the road plane (Fig. 2 (g)). In contrast to existing view synthesis methods for visualization, our approach can generate the ground-truth 2D/3D annotations, including 2D/3D bounding boxes, instance-level segmentation, and 6-DoF poses. Benefiting from the 3D vehicle template, we can further generate the dense mapping data between the image pixels and the 3D template vertices according to the 6-DoF pose. We take these dense mapping data as a bridge to perform vehicle parsing from the 2D space to the 3D space (details in Sec. 3.2).

3.1.1 Part-Based Texture Inpainting Network

From the 3D vehicle template, we propose a novel part-level texture inpainting network, which has two advantages. First, our network does not need paired multi-view images as guidance, and the input texture maps are directly projected from real-world images. Second, we use a graph-based module to learn the texture inpainting features, and this module not only encodes the individual parts but also maintains the consistency among different parts.

Fig. 3 shows our inpainting network, which adopts an encoder-decoder architecture. Specifically, in the encoder stage, our network aggregates local features from the in-
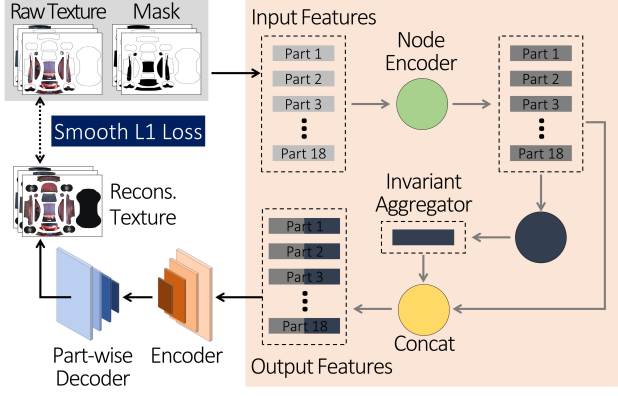
Figure 3: Our part-based texture inpainting network. We design an encoder-decoder network with the graph-based inpainting module using the smooth L1 loss.
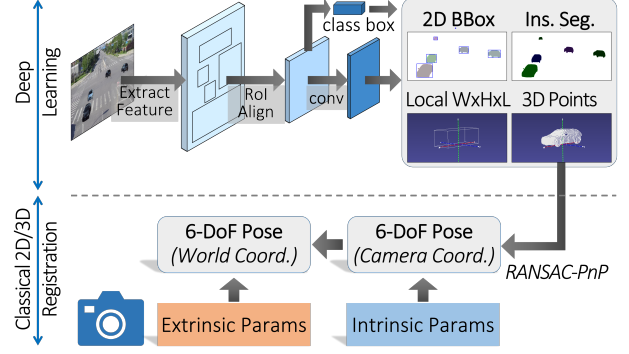

Figure 4: Robust 2D/3D vehicle parsing in different camera views using our synthesized data. We train a multi-task deep network and then utilize classical 2D/3D registration techniques to perform the 2D/3D vehicle parsing.

dividual parts and then propagates to other parts for completion. As highlighted in Sec. 3.1, the texture map P has multiple parts (*i.e.* 18) $\{p_1, p_2, ..., p_{18}\}$. Each part can be regarded as a node in the graph. Then we define a single layer of the graph propagation operation as

$$p_i^{(l+1)} = \varphi_{rel} \left\{ g_{enc} \left(p_i^{(l)} \right), \varphi_{agg} \left[g_{enc} \left(p_j^{(l)} \right) \right] \right\}, \quad (1)$$

where $p_i^{(l)}$ indicates the node feature in the l-th layer of the graph network and $p_i^{(0)}$ is the input patch of the image. The function $g_{enc} (\cdot)$ encodes the individual node/part features, $\varphi_{agg} [\cdot]$ aggregates the features of all nodes (*i.e.* 1∼18), and $\varphi_{rel} \{\cdot\}$ is the relational operator between node/part p_i and the aggregation of all nodes. In the decoder stage, we use a part-level decoder to recover each part from the last node/part feature p_i^{last}, which is defined as

$$\tilde{p}_i = g_{dec}^i \left(p_i^{last} \right). \quad (2)$$

In our implementation, $g_{enc} (\cdot)$ is a convolutional layer followed by batch normalization and then ReLU nonlinearity. $\varphi_{agg} [\cdot]$ is the part axis maxpooling. $\varphi_{rel} \{\cdot\}$ concatenates the aggregated feature to each part. $g_{dec} (\cdot)$ is a sequential deconvolutional layer to recover the i-th part image. We use a four-level part-wise graph network to encode the relations among all parts. In the training stage, we randomly mask the existing part patches to train the network to recover them in a self-supervised manner. We compute the smooth-L1 loss between p_i and $\tilde{p}_i$ for all existing parts.

3.2. Robust Approach for 2D/3D Vehicle Parsing

Estimating 3D information from a single image is a very challenging task due to projective mapping. Thanks to the synthesized data with ground-truth 6-DoF pose and real-size 3D shape (Sec. 3.1), we can perform 2D/3D vehicle parsing in different camera views. Specifically, our approach combines the strengths of deep learning and the classical 2D/3D registration techniques, which are robust for arbitrary camera intrinsic and extrinsic parameters. The key idea is producing dense 2D/3D mappings between the image pixels and the canonical 3D template vertices. More specifically, we propose a multi-task network to learn the synthesized images with ground-truth annotations. As shown in Fig. 4, our network can output the parsing results of 2D detection, instance-level segmentation, a 3D bounding box (width, height, length), and dense 3D points.

In contrast to existing "key-points"-based pose estimation approaches (*e.g.*, ApolloCar3D [45]), the key to our network is directly regressing each pixel's 3D points of the vehicle template in the canonical space. Specifically, for each pixel $c_i = (u_i, v_i)^\top$ in the region of interest (RoI), we estimate its 3D point $v_i = (x_i, y_i, z_i)^\top$ in the canonical space. This process can be formulated as

$$\tilde{V} = Regressor(C), \quad (3)$$

where $C = \{c_1, c_2, c_3, ..., c_n\}$ and $\tilde{V} = \{\tilde{v}_1, \tilde{v}_2, \tilde{v}_3, ..., \tilde{v}_n\}$. Based on the predicted 3D points, we formulate the 6-DoF pose estimation as a typical 2D/3D registration problem, which can be defined as

$$Pose = \kappa \left(\tilde{V}, C, K_{int}, K_{ext} \right), \quad (4)$$

where K_{int} and K_{ext} indicate the camera's intrinsic and extrinsic parameters, respectively. κ is the solver of the 2D/3D registration (*i.e.* Efficient-PnP or RANSAC-PnP).

In our implementation, we first use the 3D vehicle template to generate dense ground-truth mappings between C and V according to the 6-DoF poses. Then we integrate our canonical point regression module into Mask-RCNN [17] as a new branch. In addition, we add another branch to estimate the dimension $[w, h, l]$ of the vehicle. As a result, our

new multi-task network can detect/segment a vehicle and regress its 3D points and 3D dimension. The loss function of 3D points is defined as

$$L_{local3d} = SmoothL1\left(\tilde{V}, V\right),\qquad(5)$$

and the loss function of 3D dimension is defined as

$$L_{dimen} = SmoothL1\left(\left[\tilde{w}, \tilde{h}, \tilde{l}\right], [w, h, l]\right).\qquad(6)$$

3.3. CVIS Dataset

To the best of our knowledge, none of the existing datasets provide detailed 2D/3D annotation of vehicles from different views in real traffic scenarios. In this paper, we construct the first 2D/3D vehicle parsing dataset in CVIS for testing. Specifically, we capture source data from 20 street-lights in three different cities, where 1 camera and 1 Lidar are mounted at each street-light. The intrinsic and extrinsic parameters of the camera and LiDAR are pre-calibrated. Then we manually annotate the 2D bounding box and instance mask for each vehicle in the image. The associated 3D bounding box and the 6-DoF pose are labeled on the 3D point clouds. In summary, our dataset contains 14017 annotated car instances from 1540 images in different views, which are collected from the urban environment.

The size of our dataset is comparable to other datasets (*e.g.*, *CityScapes* (1525 images), *ApolloCar3D* (1041 images) for testing). Our dataset along with the source code is avaliable at *https://github.com/mh-miao/CVIS*. Note that our trained deep model is used directly for testing without any "fine-tuning" or "mix-data-training" strategies, while the training data consists of only synthesized images.

4. Experiments and Discussion

Our approach can robustly output 2D/3D parsing results under arbitrary camera parameters. Owing to the high-quality synthesized data and the efficient parsing network, our approach performs well even for long-range perception, where the vehicles have very small footprints (Fig. 5).

4.1. Implementation Details and Computation Time

We use the PerMO dataset [28] and the ApolloCar3D dataset [45] to generate our synthesized data. Specifically, PerMO provides a part-level 3D deformable vehicle template with an unfolded texture map, and ApolloCar3D labels the 6-DoF pose for each vehicle instance. The runtime for each synthesized vehicle is about 1.07 seconds. Specifically, it takes 0.07s for texture inpainting, 0.5s for background processing, and 0.5s for textured model rendering.

The training time of our network depends on the data number. In general, training 20K images costs 16 hours (8 Nvidia P40 graphics cards). In the testing phase, we directly
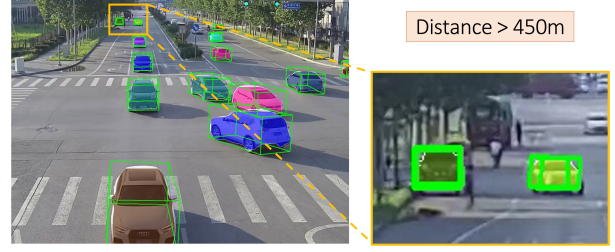

Figure 5: The 2D/3D parsing results using our approach, which is robust even for long-range perception (>450m).

use the trained model to perform 2D and 3D parsing on our dataset. For an image with 10 vehicles, the average runtime is 0.23s. Specifically, it takes 0.2s for network prediction and 0.03s for 6-DoF pose estimation.

4.2. Evaluation Metric

We use mAP as an evaluative criterion for 2D detection and instance-level segmentation because it is commonly used in many perception tasks [12].

We follow the evaluation criteria proposed in Apollo-Car3D [45] for 6-DoF pose evaluation, where *"A3DP-Abs"* (*"A3DP-Rel"*) means the absolute (relative) distance criterion. In addition, *"c-l"* (*"c-s"*) indicates results from a loose (strict) criterion. More details are introduced in [45].

4.3. Comparisons

4.3.1 Comparison with Data Augmentation Methods

To demonstrate that our synthesized data can effectively improve the network performance, we compare it with other data, including 1) *the existing dataset*, 2) *rendering data*, and 3) *GAN-based data*. Specifically, we use *KITTI* [15] as the existing dataset, which provides the vehicle annotations in the front view. We follow the approach [26] to obtain the rendering data using the 3dsMax software. The GAN-based data is generated by the approach from [29], which can synthesize novel-view vehicles according to the pre-defined poses.

To perform a fair comparison, we maintain the same number of training data to train the *Faster-RCNN* [40] network on the 2D detection task, the *Mask-RCNN* [17] network on the instance-level segmentation task, and the *DensePose* [1] network on the 6-DoF pose estimation task. As shown in Tab. 1, our synthesized data outperforms other data by a large margin. Specifically, the existing dataset only labels the front-view data, and thus the trained model is limited to other views. Rendering data has a natural domain gap with the real images, resulting in low performance. In addition, 3D rendering costs more than 5x our data augmentation approach. Although the GAN-based data look more photo-realistic, their extracted features are different from the real images in AD, limiting the network training.

Methods	2D Det. (*mAP*)	Ins. Seg. (*mAP*)	6-DoF Pose (*Abs-mean*)
Existing Dataset	49.8	48.1	22.4
Rendering Data	46.0	45.7	20.2
GAN-based Data	39.4	37.4	/
Our Data	**53.6**	**52.4**	**25.3**

Table 1: 2D/3D results with different data augmentation approaches. Note that these tasks are evaluated by SOTA methods (*i.e.* **Faster-RCNN** for 2D detection, **Mask-RCNN** for instance segmentation, and **DensePose** for 6-DoF pose).

Methods	2D Det. (*mAP*)	Ins. Seg. (*mAP*)
Pure Color	55.0	50.6
KNN Filling	57.5	51.1
Image Inpainting	55.3	47.4
Ours	**59.7**	**54.6**

Table 2: We use ***our network*** (Sec. 3.2) to quantitatively compare our inpainting method with other inpainting methods on the tasks of 2D detection and instance segmentation.

4.3.2 Comparison with Hole-Filling Methods

To justify the effectiveness of our part-based texture inpainting approach, the following hole-filling methods are compared. 1) *Pure Color Filling.* We directly fill the missing regions using a pure color, which is a mean value of the existing region. 2) *KNN Filling.* We fill the missing regions using a linear blending algorithm where each pixel value is determined by the K-nearest neighbors (KNN). 3) *Image-based Inpainting.* We consider each vehicle instance, remove the complete area randomly, and use the information of the remaining area to train an encoder-decoder network to inpaint the removed area. The removed area can be regarded as ground-truth for the training network. 4) *Ours: Part-based Texture Inpainting.* We train a part-based inpainting network on the texture map. Here, we use our network (Sec. 3.2) to train these data. Tab. 2 shows that our approach outperforms the second-best results on 2D detection and instance segmentation by 2.2% and 3.5%, respectively. The filling results of different approaches are shown in Fig. 6. Our approach can inpaint the missing regions while maintaining the color and structure consistency.

4.3.3 Robust Performance on 6-DoF Pose Estimation

Existing pose estimation methods can be categorized into three classes: 1) *"Direct" methods*, which directly regress the vehicle poses from the 2D images (*e.g.*, 3D-RCNN [23]); 2) *"Key-points"-based methods*, which ex-
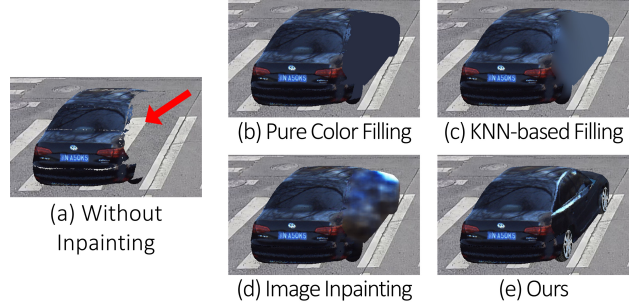

(a) Without Inpainting

(b) Pure Color Filling

(c) KNN-based Filling

(d) Image Inpainting

(e) Ours

Figure 6: Visualization results of our part-based inpainting approach and other hole-filling methods.

Methods	A3DP-Abs			A3DP-Rel		
	mean	*c-l*	*c-s*	*mean*	*c-l*	*c-s*
ApolloCar3D	23.3	31.7	26.7	22.5	28.7	23.8
DensePose	25.3	35.6	28.7	25.1	32.7	26.7
Our Method	**33.0**	**47.5**	**38.6**	**27.8**	**38.6**	**29.7**

Table 3: 6-DoF pose evaluation with different approaches. We introduce the evaluation metric in Sec. 4.2.

tract the pre-defined key-points and then solve a PnP problem to obtain the 6-DoF pose (*e.g.*, DeepMANTA [4], ApolloCar3D [45]); and 3) *"Dense mapping"-based methods*, which regress the dense mapping between the 2D image and the UV map of the 3D model before solving the 6-DoF pose. (*e.g.*, DensePose [1]). As reported by [45], ApolloCar3D advances 3D-RCNN and DeepMANTA by a big margin. Therefore, we retrain the ApolloCar3D network and the DensePose network using the same training data for fair comparison. Tab. 3 shows the 6-DoF pose estimation results. Our approach outperforms the ApolloCar3D and DensePose on "Abs-mean" by 9.7% and 7.7%, respectively.

4.3.4 Robust Performance on 3D Detection

To justify the robustness of our 2D/3D parsing approach, we generate images with different intrinsic parameters of the camera while maintaining the same camera extrinsic parameters and vehicles' poses (Fig. 7). Many SOTA methods (*e.g.*, [30], [3], [9]) fail in such cases because they are trained by the depth with fixed camera parameters. As a result, they predict these vehicles with varying poses, which follow perspective projection (big for near and small for far). In contrast, our method outputs the canonical 3D points which are one-2-one mapped to image pixels, and then computes the 6-DoF poses using the RANSAC-PnP algorithm. Fig. 7 shows that our 3D detection results are accurate compared to the ground-truth.

Figure 7: 3D detection results in bird's eye view with different intrinsic parameters. The red box indicates the ground-truth, and the boxes in other colors indicate our results. Our detection results are very close to the ground-truth in the 3D space, though the visual 2D images are different.

4.4. Performance Analysis

4.4.1 The Impact of Vehicle Shadow

As highlighted in Sec. 3.1, generating a vehicle's shadow is important for improving the quality of synthesized images. Here, we train our network using the synthesized data with and without shadows. Tab. 4 lists the evaluation results. The data synthesized with shadows outperforms the other data on the tasks of 2D detection, instance segmentation, and 6-DoF pose by 1.4%, 1.5%, and 2.1%, respectively.

Tasks	w/o Shadow	with Shadow
2D Detection (mAP)	58.3	**59.7**
Ins. Seg. (mAP)	53.1	**54.6**
6-DoF Pose (Abs-mean)	30.9	**33.0**
6-DoF Pose (Rel-mean)	25.9	**27.8**

Table 4: Ablation study of shadows on 2D/3D tasks. The data synthesized with shadows can effectively improve the performance of 2D/3D parsing.

4.4.2 Diversity of the Synthesized Data

We highlight that we can generate diverse data. As shown in Fig. 8, we can synthesize photo-realistic texture maps from the real images. These texture maps are mapped to a deformable vehicle template (*e.g.*, [28]), which can be deformed to many vehicle models with different types (*e.g.*, car, SUV, MPV, *etc*.). Finally, we render these textured models with different intrinsic and extrinsic parameters of cameras under various illuminations.

4.4.3 Improvements on the KITTI Dataset

In this paper, we synthesize images for vehicle perception in CVIS. To justify the effectiveness and generalizability of
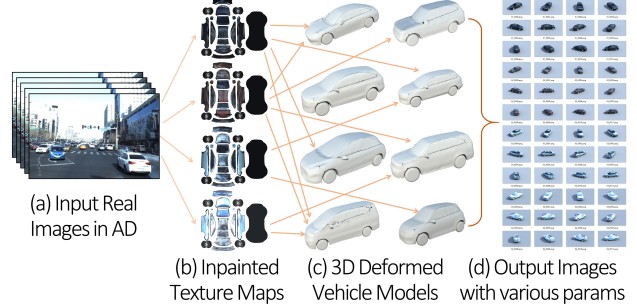

Figure 8: We use the real-traffic images for texture generation and the deformable 3D template for model generation, plus various illuminations for 3D rendering. All of the factors ensure the diversity and fidelity of our synthesized data.

our approach, we evaluate it on the *KITTI* dataset [15, 37]. Specifically, we mix our synthesized data of CVIS (2593 images) with the existing training data of *KITTI* (7481 images) to train *Mask-RCNN* [17]. As shown in Tab. 5, the *mAP* values of 2D detection and instance segmentation are improved by 1.4% and 1.5%, respectively.

Methods	2D Det.	Ins. Seg.
Existing Training Data	43.7	39.4
Mix Data (Existing + Ours)	**45.1**	**40.9**

Table 5: Our synthesized data can also improve the 2D detection and instance segmentation performance on *KITTI*. The *mAP* values are the higher the better.

5. Conclusions

In this paper, we present a novel approach for 2D/3D vehicle parsing corresponding to novel views for CVIS. Instead of manually annotating data for network training, we propose a part-assisted view synthesis approach for data augmentation, which can automatically synthesize novel-view images with ground-truth 2D/3D annotations. Moreover, we present a new approach that combines deep learning and classical 2D/3D registration techniques to perform robust 2D/3D parsing of vehicles, which can be applied to distinct camera parameters. As part of developing these algorithms for CVIS, we will be able to re-utilize the capabilities of prior AD datasets in a novel manner.

Acknowledgement

References

[1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, pages 7297–7306, 2018.

[2] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, pages 8340–8348, 2018.

[3] Yingjie Cai, Buyu Li, Zeyu Jiao, Hongsheng Li, Xingyu Zeng, and Xiaogang Wang. Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. In *AAAI*, 2020.

[4] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *CVPR*, 2017.

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, et al. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015.

[6] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 3-sweep: Extracting editable objects from a single photo. *ACM TOG*, 32(6):1–10, 2013.

[7] Marius Cordts, Mohamed Omran, et al. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[8] Angela Dai, Angel X. Chang, et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[9] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR Workshops*, pages 1000–1001, 2020.

[10] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *ICCV*, pages 1301–1310, 2017.

[11] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring spatial context for 3d semantic segmentation of point clouds. In *ICCV*, pages 716–724, 2017.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[13] Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. Cityflow-nl: Tracking and retrieval of vehicles at city scale by natural language descriptions. 2021.

[14] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[18] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *CVPR Workshops*, pages 954–960, 2018.

[19] Biliana Kaneva, Antonio Torralba, and William T Freeman. Evaluation of image features using a photorealistic virtual world. In *ICCV*, pages 2282–2289. IEEE, 2011.

[20] Amlan Kar, Aayush Prakash, Ming-Yu Liu, et al. Meta-sim: Learning to generate synthetic datasets. In *CVPR*, pages 4551–4560, 2019.

[21] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM TOG*, 30(6):1–12, 2011.

[22] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM TOG*, 33(4):1–12, 2014.

[23] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, pages 3559–3568, 2018.

[24] Xinchen Liu, Wu Liu, et al. Beyond the parts: Learning multi-view cross-part correlation for vehicle re-identification. In *ACM MM*, 2020.

[25] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European conference on computer vision*, pages 869–884. Springer, 2016.

[26] Zongdai Liu, Feixiang Lu, Peng Wang, Hui Miao, Liangjun Zhang, Ruigang Yang, and Bin Zhou. 3d part guided image editing for fine-grained object understanding. In *CVPR*, pages 11336–11345, 2020.

[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM TOG*, 34(6):1–16, 2015.

[28] Feixiang Lu, Zongdai Liu, Xibin Song, Dingfu Zhou, Wei Li, Hui Miao, Miao Liao, Liangjun Zhang, Bin Zhou, Ruigang Yang, and Dinesh Manocha. Permo: Perceiving more at once from a single image for autonomous driving. *arXiv preprint arXiv:2007.08116*, 2020.

[29] Kai Lv, Hao Sheng, Zhang Xiong, Wei Li, and Liang Zheng. Pose-based view synthesis for vehicles: A perspective aware method. *IEEE Transactions on Image Processing*, 29:5163–5174, 2020.

[30] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *ICCV*, pages 6851–6860, 2019.

[31] Dechao Meng, Liang Li, et al. Parsing-based view-aware embedding network for vehicle re-identification. In *CVPR*, pages 7103–7112, 2020.

[32] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, pages 909–918, 2019.

[33] Milind Naphade, Rama Chellappa, David Anastasiu, et al. The 5th ai city challenge. In *CVPR Workshops*, 2021.

[34] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.

[35] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *CVPR*, pages 3500–3509, 2017.

[36] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, pages 4561–4570, 2019.

[37] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019.

[38] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[39] Konstantinos Rematas, Chuong H Nguyen, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Novel views of objects from a single image. *IEEE TPAMI*, 39(8), 2016.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[41] Reportlinker.com. Cooperative vehicle infrastructure system (cvis) and vehicle to everything (v2x) industry report. In *https://www.reportlinker.com/p05627064*, pages 1–165, 2018.

[42] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1297–1304, 2011.

[43] Joshua E Siegel, Dylan C Erb, and Sanjay E Sarma. A survey of the connected vehicle landscape—architectures, enabling technologies, applications, and development areas. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2391–2406, 2017.

[44] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *NeurIPS*, 2019.

[45] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *CVPR*, pages 5452–5462, 2019.

[46] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, pages 2686–2694, 2015.

[47] Zheng Tang, Milind Naphade, Stan Birchfield, et al. Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In *ICCV*, pages 211–220, 2019.

[48] Zheng Tang, Milind Naphade, Ming-Yu Liu, et al. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *CVPR*, June 2019.

[49] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, pages 23–30. IEEE, 2017.

[50] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[51] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, pages 8798–8807, 2018.

[52] Chen Xin, Yuwei Li, Xi Luo, Tianjia Shao, Jingyi Yu, Kun Zhou, and Youyi Zheng. Autosweep: Recovering 3d editable objects from a single photograph. *IEEE TVCG*, 2018.

[53] Xiaogang Xu, Ying-Cong Chen, and Jiaya Jia. View independent generative adversarial network for novel view synthesis. In *ICCV*, pages 7791–7800, 2019.

[54] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM TOG*, 38(4):1–13, 2019.

[55] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NeurIPS*, pages 1099–1107, 2015.

[56] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, pages 3973–3981, 2015.

[57] Shunyu Yao, Tzu Ming Harry Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, William T Freeman, and Joshua B Tenenbaum. 3d-aware scene manipulation via inverse graphics. *arXiv preprint arXiv:1808.09351*, 2018.

[58] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating content consistent vehicle datasets with attribute descent. In *ECCV*, August 2020.

[59] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3d objects with differentiable rendering of sdf shape priors. In *CVPR*, 2020.

[60] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *ICCV*, pages 1941–1950, 2019.

[61] Rong Zhang, Wei Li, Peng Wang, Chenye Guan, Jin Fang, Yuhang Song, Jinhui Yu, Baoquan Chen, Weiwei Xu, and Ruigang Yang. Autoremover: Automatic object removal for autonomous driving videos. In *AAAI*, volume 34, pages 12853–12861, 2020.

[62] Youyi Zheng, Xiang Chen, Ming-Ming Cheng, Kun Zhou, Shi-Min Hu, and Niloy J Mitra. Interactive images: cuboid proxies for smart image manipulation. *ACM TOG*, 31(4):99–1, 2012.

[63] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, pages 286–301. Springer, 2016.

[64] Hao Zhu, Hao Su, Peng Wang, Xun Cao, and Ruigang Yang. View extrapolation of human body from a single image. In *CVPR*, pages 4450–4459, 2018.

[65] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.