Pose-Based View Synthesis for Vehicles: A Perspective Aware Method

Kai Lv, Hao Sheng, *Member, IEEE*, Zhang Xiong, Wei Li, *Member, IEEE*, and Liang Zheng, *Member, IEEE*

Abstract— In this paper, we focus on the problem of novel view synthesis for vehicles. Some previous works solve the problem of novel view synthesis in a controlled 3D environment by exploiting additional 3D details (*i.e.*, camera viewpoints and underlying 3D models). However, in real scenarios, the 3D details are difficult to obtain. In this case, we find that introducing vehicle pose to represent the views of vehicles is an alternative paradigm to solve the lack of 3D details. In novel view synthesis, preserving local details is one of the most challenging problems. To address this problem, we propose a perspective-aware generative model (PAGM). We are motivated by the prior that vehicles are made of quadrilateral planes. Preserving these rigid planes during image generation ensures that image details are kept. To this end, a classic image transformation method is leveraged, *i.e.*, perspective transformation. In our GAN-based system, the perspective transformation is applied to the encoder feature maps, and the resulting maps are regarded as new conditions for the decoder. This strategy preserves the quadrilateral planes all the way through the network, thus shuttling the texture details from the input image to the generated image. In the experiments, we show that PAGM can generate high-quality vehicle images with fine details. Quantitatively, our method is superior to several competing approaches employing either GAN or the perspective transformation. Code is available at: https://github.com/ilvkai/view-synthesis-for-vehicles

Index Terms— Novel view synthesis, generative adversarial nets, perspective transformation, generative model, vehicle pose.

I. INTRODUCTION

IN THIS work, we study the problem of novel view synthesis for vehicles in real scenarios. Given an input vehicle image, we aim to synthesize new images of the same vehicle from another view. Synthesizing novel views for vehicles has a variety of practical applications in computer vision and virtual reality. Vehicle novel view synthesis can be regarded as a data augmentation method and contribute to the vehicle re-identification task [1], [2]. On the one hand, vehicle generation can be applied as a data augmentation method. The generated images have been successfully exploited for training deep learning frameworks for relevant recognition and re-identification tasks [3]–[5]. On the other hand, vehicle generation can help improve the discriminative capability and robustness of the ReID models [6]. Meanwhile, generating vehicles from different viewpoints is also significant to some practical applications, *i.e.*, virtual reality. For example, it enables photo editing programs to manipulate objects in 3D rather than 2D. Also, it could help create full virtual reality environments based on the objects with desired viewpoints or poses [7].

To address the view synthesis task, much effort has been made in 3D-based methods, which use the underlying 3D models and camera viewpoints.) Some geometry-based methods [9]–[11] model the underlying 3D geometry and benefit from implicit or explicit geometric reasoning. These methods depend on the input view and might fail under occlusion. Other methods take advantage of appearance flow [12]–[14] to move the pixels from the input image to novel views. Appearance flow is represented by a set of 2D coordinate vectors specifying which pixels in the input view could be used to reconstruct the target view. However, the 3D details, such as the underlying 3D models or camera viewpoints, are usually not available in the real-world scenarios. This obstacle compromises the application scope of the 3D-based methods.

In this paper, we investigate vehicle pose to address the lack of 3D information in real scenarios. Following the Stacked Hourglass Networks [15], we extract the vehicle pose from an image containing a vehicle. The vehicle pose is represented by a set of 2D keypoints. In this work, we define vehicle view synthesis as follows: given a vehicle image and a specific pose, we propose to automatically generate an image containing the same vehicle with the specified pose. Note that due to the introduction of vehicle pose, the novel view synthesis for vehicles can be regarded as vehicle pose transfer. The main

Fig. 1. Examples of vehicle images generated by different methods. **Column 1:** source images. **Column 2:** specified pose / orientation. **Column 3 - 5:** images generated by perspective transformation, Conditional GAN [8] and our method, respectively. Incorporating both perspective transformation and GAN, our method can generate images that 1) are realistic and 2) have preserved texture details (*e.g.,* the license plate and logo).
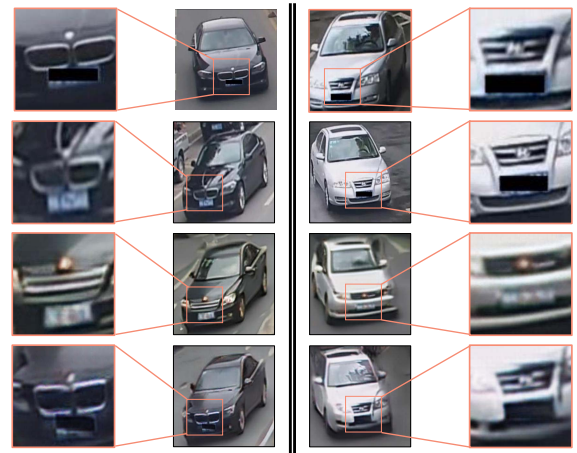

Fig. 2. A closer look at the vehicle details. Row 1 and Row 2: input images and target ground truths, respectively. Row 3 and Row 4: generated images by CGAN [8] and our method, respectively. On the two side columns, enlarged views of license plates and logos are shown. Comparing with CGAN, our method is closer to the ground truth and better at preserving local details.

reason is that vehicles are rigid objects, with different poses implying different views.

The main challenge of vehicle view synthesis lies in generating images that maintain the integrity of vehicles while also preserving texture details, such as the texture of rigid license plates and logos. Two basic techniques can be used to solve the problem, *i.e.*, perspective transformation and generative models. The former is based on pure geometry [16]. It assumes a 3D model of a vehicle and projects each plane from one perspective in the input image to another in the target image. Note that 3D information is not used in our method. Due to its geometric nature, the projected planes preserve the texture details very well. The generative model, on the other hand, is based on the generative adversarial network (GAN) [17]. The generated vehicle images should follow the same distribution as the real images.

From two aspects, the prior art motivates us to design this work. First, perspective transformation is advantageous in preserving image texture details, but fails to "imagine" the non-existing parts, so the generated images look fake. In Column 3 of Fig. 1, sample images synthesized by the perspective transformation are shown. Because the source image patches are directly shuttled to the target location, we observe very clear details. However, the inevitable consequence is that the vehicles are incomplete, *i.e.*, some invisible source parts remain invisible in the target image. Moreover, the background is not attended to, either. Therefore, in spite of its ability to preserve details, merely using the perspective transformation is not effective in generating realistic vehicle images.

Second, GAN-based methods can generate realistic images, but some texture details of the generated images may change significantly. In Column 4 of Fig. 1 and Row 3 of Fig. 2, we show images generated by the conditional generative adversarial network (CGAN) [8]. Because GAN is optimized on some training set, the generated images are, to some extent, biased towards the training data. For example, the GAN generated images usually have a visible license plate, because the license plates of most training images are visible. In this

case, when the source image has a masked license plate, the generated one may deviate much from it. From these examples, we also observe that the generated logo and the overall car shape can be very different from the source image. This problem severely limits the application of GAN in rigid object generation.

This paper aims to generate vehicle images that 1) are realistic, and 2) have preserved texture details. Considering the complementary properties of the perspective transformation and GAN, a natural idea is to leverage their respective strengths. We thus propose the perspective-aware generative model (PAGM). In its structure, PAGM takes advantage of both the perspective transformation and GAN. On the one hand, it contains a generator and a discriminator for realistic image generation. More precisely, the generator consists of an encoder and a decoder. On the other hand, we design several perspective transformation modules attached to the convolutional layers of the encoder. These modules perform perspective transformation on the encoder feature maps, and the vehicle planes are shuttled to their corresponding spatial positions without information loss. The resulting feature maps are concatenated to the decoder feature maps, serving as conditions to preserve texture details in the generated images. In summary, this paper makes the following main points.

- We are among the first to render pose-specified vehicles in real scenarios, providing insight into rigid object view synthesis.
- PAGM not only generates realistic vehicles, but importantly, preserves the texture details, *i.e.*, we can generate vehicles that look realistic and preserve source texture information.
- We provide insightful analyses of the components, *i.e.,* conditional GAN, skip connections, Perspective Transformed Connections (PTC) and perceptual loss.
- Experiment confirms the effectiveness of PAGM qualitatively and quantitatively.

II. Related Work

A. Generic Image Generation

Generative adversarial nets (GAN) [17] have achieved impressive success in image generation in recent years and there exist many image generation applications [12], [18]–[22]. GAN is a min-max two-player game containing a generator G and a discriminator D. G tries to produce fake images that the D cannot figure out whether these images are real or fake. However, the images generated by GAN [17] are determined by a random vector and the results cannot be controlled. To overcome this problem, a conditional version named conditional adversarial nets (CGAN) [8] is proposed. CGAN conditions not only the generator but also the discriminator by introducing external cues. In this paper, the proposed method mainly adopts the CGAN framework. Moreover, image-to-image translation translates an image from the source domain to the target domain [18], [23]. Zhu *et al.* [24] propose the CycleGAN, which has a cycle consistent loss to learns a forward mapping and an inverse mapping. Sung *et al.* [25] propose to generate images from scene graphs. This method enables explicitly reasoning about objects and their relationships. Using the given conditions, Gauthier [26] are able to train models to generate faces with specific attributes.

B. Novel View Synthesis

The methods in novel view synthesis [12], [14], [22], [27]–[32] mainly fall into two categories: *geometry-based approaches* and *appearance flow approaches*. Geometry-based approaches benefit from geometric reasoning in solving the view synthesis problems. Furukawa *et al.* [28] propose to use multiple images when explicitly reconstructing the 3D scene, which is then used to synthesize novel views. Garg *et al.* [29] first predict a depth map and then propose to transform each reconstructed 3D point in the depth map when synthesizing novel view images. Optical flow [33] can be utilized in the view synthesis task by providing dense pixel-to-pixel correspondence between two images. Zhou and Shao [12] propose to use appearance flow to solve the task of novel view synthesis. In [12], the authors map the pixels in the source view to the target view. Based on appearance flow, Park *et al.* [14] add an image completion network to deal with occlusion and disocclusion. Sun *et al.* [13] propose an end-to-end trainable framework based on images of multiple viewpoints. In [13], the model does not require 3D supervision, but the camera pose is needed to predict a dense flow field. Recently, Palazzi *et al.* [34] generate novel views of objects in a semi-parametric setting: relying on both 3D CAD models and an image completion network. These previous methods can be utilized to solve vehicle view synthesis in a controlled 3D environment and require underlying 3D models or camera viewpoints. However, our method does not utilize the 3D models/camera viewpoints and merely takes the pose extracted from images as inputs. In this sense, our method would be more advantageous in the real world, because poses can be conveniently obtained using off-the-shelf pose extractors.

C. Person Pose Transfer

Some recent works discuss person pose transfer, which takes person pose as input. For example, Zheng *et al.* [3] use GAN [17] to generate unlabeled person samples to regularize the person re-identification fine-tuning. By combining VAE [35] and GAN [17] together, Lassner *et al.* [36] present a model to generate images of a person with different clothes. Zhao *et al.* [37] generate multi-view cloth images from a single view cloth image by adopting a coarse to fine method. Ma *et al.* propose to synthesize person images into a target pose. The input is an image of the same person and the target pose [38]. Ma *et al.* later propose a two-stage reconstruction pipeline that learns a disentangled representation of the aforementioned image factors and generates novel person images at the same time [39]. Generally, we follow the setting of [38]–[40]. we aim to generate pose-specified objects using a generative model. The closest work to ours is [40]. In [40], the affine transformation is used to shuttle rectangle patches from the source to the target image. The difference between this work and [40] comes from our focus which is on rigid objects, *i.e.*, vehicles. For a human body, most existing works attempt to generate images with well-preserved texture details, *i.e.*, texture of non-rigid clothes and faces. Different from persons, vehicles are rigid objects and have different types of texture, *i.e.*, texture of rigid license plates and logos. Moreover, vehicles can be decomposed into irregular quadrilateral planes, instead of regular rectangles, allowing us to explore a much more elaborated way (perspective transformation) to project these planes. In experiment, we show that our method is significantly superior to the affine transformation method [40].

D. Vehicle Image Generation

There exist some vehicle image generation methods designed for vehicle-based applications like fine-grained image classification [16], [41], [42] and vehicle re-identification [43]–[46]. Zhou *et al.* [1] learn the features of a vehicle captured and take the features as conditional variables to effectively generate cross-view images to contribute to vehicle re-identification. An inspiring work for ours is [16]. In [16], Sochor *et al.* attempt to preprocess the vehicle images for better alignment to improve fine-grained classification. They use 3D vehicle bounding boxes and do patch alignment with an affine warp to "unpack" a vehicle image. The unpacked image does not look like a car but benefits classification, because it localizes the vehicle parts and normalizes their positions. The similarity between PAGM and [16] is that both exploit the 3D layout of vehicles. The difference is that we aim to improve vehicle generation, while [16] consider classification.

III. Proposed Approach

In Section III-A, We first describe the overall system (Fig. 3), which is based on CGAN. We then describe vehicle pose estimation and the definition of quadrilateral regions in III-B. We introduce PAGM in Section III-C and provide in-depth discussions of PAGM in Section III-D.
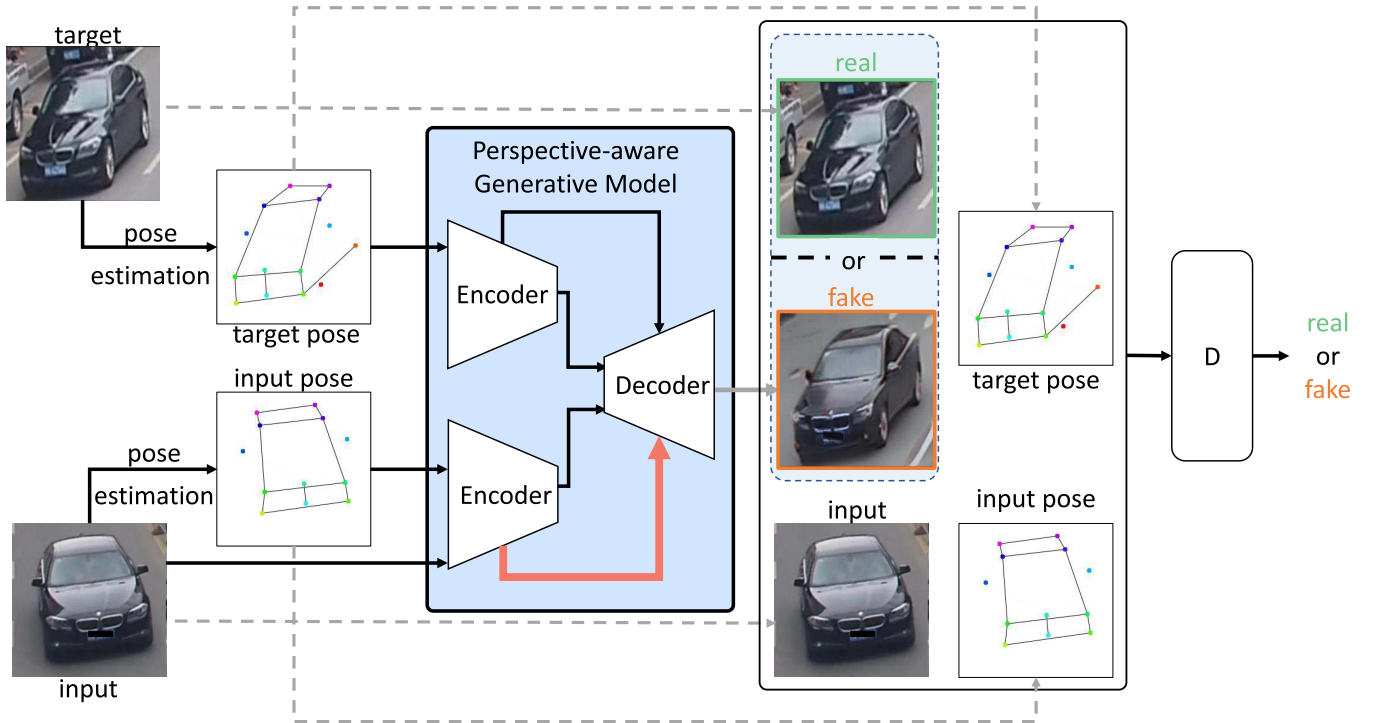
Fig. 3. Overall system workflow. Our system has two major components: the proposed perspective-aware generative model (PAGM) and a discriminator (D). In training, a pair of real images as well as their pose maps are input to the framework. The two images consist of a source image and an image with the target pose. Pose response maps are calculated by an off-the-shelf pose estimator [15]. The contribution of this paper is the PAGM component, which is essentially a generator. PAGM is described in Section III-C and Fig. 5.

A. System Workflow

1) Input and Output: In training, a pair of images, a source image I_s and a target image I_t, are input to the system. The first step is to extract the vehicle poses: P_s and P_t. Then, the poses are converted to response maps R_s and R_t R, which are the input of the GAN-based model. Specially, the input of PAGM can be defined as (R_t, I_s, R_s). Then PAGM outputs a fake image $\hat{I}$ which will be fed into a discriminator. Thus, the discriminator takes (I, R_t, I_s, R_s) where I is a real image I_t or a fake one $\hat{I}$. The discriminator finally outputs a scale to judge whether I is real or fake.

2) CGAN: Based on GAN [17], CGAN [8] is proposed by feeding extra information y into both the discriminator and the generator. For our method, the conditions y can be defined as (R_t, I_s, R_s). Then, the adversarial loss of CGAN can be written as,

$$\mathcal{L}_{CGAN}(D, G) = E_{I_t \sim p_{data}(I_t)}[\log D(I_t|y)]$$
$$+ E_{z \sim p_z(z)}[\log(1 - D(G(z|y)))]. \quad (1)$$

Apart from $\mathcal{L}_{CGAN}$, L_1 and L_2 loss could also be used in PAGM. As mentioned in [18], the L_2 loss might result in blurring, which is undesirable. Thus, in this work, L_1 loss is adopted and defined as,

$$\mathcal{L}_{L_1}(I_t, \hat{I}) = ||I_t - \hat{I}||_1, \quad (2)$$

where $\hat{I}$ is the images generated and I_t is the target image.

Inspired by [19], we also apply the perceptual loss $\mathcal{L}_{perceptual}$ to enforce image structure similarity between the generated image and the target image. In implementation, we utilize the VGG19 [47] model pretrained on ImageNet to extract the multi-level feature maps. $\phi_j(I)$ means the feature at the jth layer of the VGG19 network for the image I. The perceptual loss $\mathcal{L}_{perceptual}$ is defined as,

$$\mathcal{L}_{perceptual}(\hat{I}, I_t) = \frac{1}{C_j H_j W_j}||\phi_j(\hat{I}) - \phi_j(I_t)||_2^2, \quad (3)$$

where $\phi_j(I)$ is the activations of the jth layer of the network ϕ when processing the image I. The shape of the feature map $\phi_j(I)$ is $C_j \times H_j \times W_j$.

3) Overall Objective Function: We combine L_1, $L_{perceptual}$ and $\mathcal{L}_{CGAN}(D, G)$ to obtain the final objective function,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{L_1} + \lambda_2 \mathcal{L}_{perceptual} + \mathcal{L}_{CGAN}, \quad (4)$$

where λ_1 and λ_2 control the relative importance of the losses.

B. Vehicle Pose Estimation and Region Definition

1) Pose Estimation: To generate a pose-specified object, the first and foremost step is to define and extract poses. It is also performed in pose-based person generation [25], [38], [39]. In a similar spirit to human pose, $p = 20$ keypoints [45] are annotated for the VeRi-776 dataset [43], [44] to depict the skeleton of vehicle. Vehicle keypoints are defined at some discriminative locations. In this paper, given an image $I \in \mathbb{R}^{3 \times h \times w}$, we use the Stacked Hourglass Networks [15] to predict the vehicle keypoints $P(I)$. The keypoints are 2D coordinates and are denoted as $P(I) = \{(x_1, y_1), (x_2, y_2), \cdots, (x_p, y_p)\}$. In pose estimation, the keypoints $P(I)$ are converted to a set of Gaussian response maps $R(P(I))$, which are the input of PAGM and the discriminator.
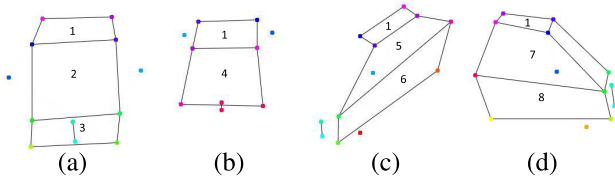
Fig. 4. The quadrilateral planes and keypoints of a vehicle. The orientation of the four vehicles is front, back, left and right. Note that the regions are usually irregular quadrilaterals. (1) roof, (2) front windshield, (3) front logo area, (4) back windshield, (5) left windows, (6) left body, (7) right windows and (8) right body.

The Gaussian maps consist of p channels and each channel can be written as,

$$R(l_i) = exp\left(-\frac{\|l - l_i(I)\|}{\sigma^2}\right), \tag{5}$$

where l_i is the location of i-th keypoint and $\sigma = 6$ pixels.

2) Quadrilateral Regions: Given the keypoints, we manually cut a vehicle into $r = 8$ quadrilaterals (see Fig. 4). Each region is defined by 4 keypoints. The quadrilaterals contain most of the local details of vehicles like logo, lamp, and plate. Note that these regions are irregular quadrilaterals rather than parallelograms.

C. Adversarial Perspective Vehicle Generation

1) Perspective Transformed Connections: In order to generate detail-preserving vehicle planes, this paper proposes a perspective transformation module, named perspective transformed connections (PTC). It works on the encoder feature maps, and the resulting feature maps are used as the conditions for the decoder.

Before describing PTC, we first review the perspective transformation (also called the projective transformation) and the affine transformation. The perspective transformation is a geometric operation that projects the planes in one perspective to another by multiplying a matrix $T = \begin{pmatrix} a_1 & a_2 & b_1 \\ a_3 & a_3 & b_2 \\ c_1 & c_2 & 1 \end{pmatrix}$, where $\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ is a rotation matrix. This matrix defines the transformation that an image will be undertaken: scaling, rotation, etc. In this matrix, $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ is the translation vector, and $\begin{pmatrix} c_1 & c_2 \end{pmatrix}$ is the projection vector. The affine transformation is a particular case of the perspective transformation and its matrix A can be written as $A = \begin{pmatrix} a_1 & a_2 & b_1 \\ a_3 & a_3 & b_2 \end{pmatrix}$.

In [48], the affine transformation is used to process the feature maps. In this paper, we use the perspective transformation instead. The reason is that affine transformation can only deal with parallelograms: it projects parallelograms into parallelograms. In comparison, the perspective transformation can project an irregular quadrilateral into another quadrilateral. Because each vehicle is divided into irregular quadrilaterals, only perspective transformation can be applied. This is a fundamental difference between this work and [40]. The perspective transformation is defined as,

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \times T = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}, \tag{6}$$

where (x, y) and (x', y') denote a pixel in the source image and the target image, respectively.

Then, we describe PTC, which is based on perspective transformation. PTC takes the feature maps of encoder as input and output perspective transformed feature maps. As the vehicles are decomposed into r regions, r perspective transformations are required to shuttle the source to the target. For the i-th region, a perspective transformation matrix T_i and a mask M_i is required. In order to get T_i, two quadrilaterals are required to compute the translation from source to target. The mask M_i has the same size with target vehicle image and the locations in i-th region are labeled with 1. Note that as a vehicle is a rigid object, it is impossible to view all the r vehicle planes. If the i-th region is occluded, the corresponding mask is a matrix $\mathbf{0}$. The i-th perspective transformed connections F_t then is defined as,

$$F_i = T_i(F_s) \odot M, \tag{7}$$

where F_s is the feature map fetched from encoder and $\odot$ means a point-wise multiplication. As the r regions do not have overlaps, the feature maps of each region can be added directly. Then the final perspective transformed feature maps F can be defined as,

$$F = \sum_{i=1}^{r} F_i. \tag{8}$$

Similar to the images generated by perspective transformation (Fig. 1), PTC also has some zero areas. Note that PTC only handles the planes which have appeared in source image and target image. Finally, the transformed feature maps are sent to the decoder layers and concatenated with the feature maps of decoder.

2) Two-Way Skip Connections: The two-way skip connections locate between the two encoders and the decoder. Given the specific perspective transformed feature maps from source vehicle encoder, the decoder also has connections from the target pose encoder. As described above, the R_s or R_t has p-channel response maps that have the same size with the input image. We introduce two encoders to extract two vectors as the conditions of the decoder. More specially, the input image I_s and source pose R_s are concatenated together and pass through an encoder network. Meanwhile, the R_t is fed into the other encoder. Note that the two nets have the same structure but do not share weights.

D. Discussions

1) What Is the Relationship Between View Synthesis and Pose Transfer? : On the one hand, human pose transfer does not mean view synthesis. Human pose refers to the positions of human joints and is represented by a set of coordinates. As a human body is a non-rigid object, different poses always imply that the relative positions of the joints are not consistent. Meanwhile, different views refer that the view of the observer has changed and the relative positions of joints should keep the same. Thus, we argue that human pose transfer is different from the novel view synthesis task. On the other hand, vehicle pose transfer can be regarded as view synthesis for vehicles.
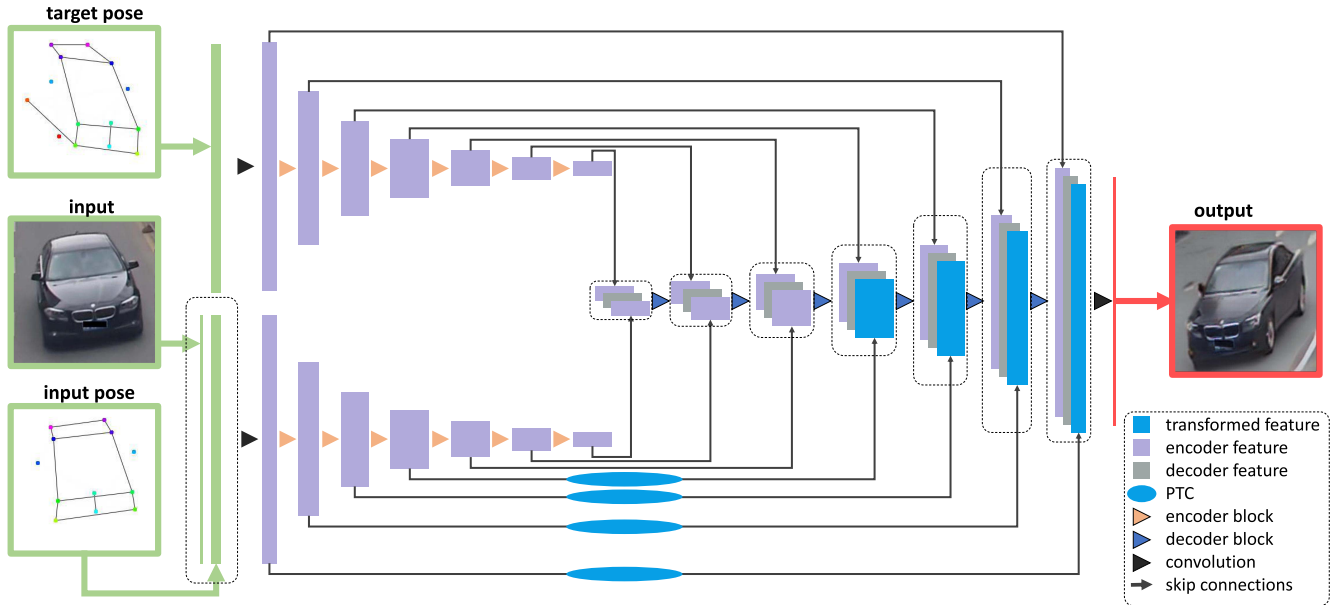
Fig. 5. Proposed perspective-aware generative model (PAGM). We have three inputs (in green boxes): a source image, its pose and a target pose. The output (in red box) is a generated vehicle image. There are two encoders to provide conditions and one decoder for image generation. The first encoder takes source image (bottom left) and its pose (top left) as input. The second encoder only takes target pose (bottom left) as input. Two-way skip connections (black arrows) shuttle details from encoder layers to decoder layers. Four perspective transformation modules (blue oval) are used to process feature maps from the first encoder and produce perspective transformed feature maps (blue rectangle).

Unlike a human body, a vehicle is a rigid object, whose different poses are the same with different views.

2) Why Doesn't Conditional GAN Preserve Local Details?: In CGAN, the image details are somewhat lost during encoding. The output of the encoder is a vector, whose dimension is too low to maintain all the details of the input image. Then, the vector is input to the decoder as conditions that largely express some global features (*e.g.,* the color). Meanwhile, note that the pose condition is preserved well. The main reason may be that the pose of a vehicle can be represented by p keypoints and the conditional vector is sufficient to include the pose clues.

3) Why Don't U-Net Skip Connections Work?: U-net [49] is first introduced to solve the image segmentation problem, where images of source and target domain are pixel-to-pixel aligned. The feature maps of the encoder, which have the same spatial distribution with the ones of the decoder, are strong constraints while the misaligned ones are not qualified for strong constraints. However, in our problem, as the input and target images have different poses, the feature maps would be misaligned. Thus, directly applying U-net skip connections only provides the decoder with misaligned feature maps and contributes little to this work.

4) Why Does PTC Work?: On the one hand, instead of delivering the decoder misaligned feature maps, PTC takes the misaligned feature maps as input and aligns the maps by using perspective transformation. The processed feature are then sent to the decoder and have the same spatially distribution with the feature maps of the decoder in terms of the details. By adopting PTC, the pixel-to-pixel misaligned task (input and target have different poses) is converted to the pixel-to-pixel aligned task to some extent. On the other hand, the PTC also filters out the background by the region masks. As the

background is useless, filtering out the background makes the decoder more focus on generating details.

IV. EXPERIMENT

A. Dataset and Evaluation Protocol

There exist several datasets related to vehicle-based applications, e.g, VeRi-776 [43], [44], VehicleID [50], Box-Cars21k [16], and CompCars [41]. In this paper, we conduct the experiment on VeRi-776 and VehicleID, which are mainly used for vehicle re-identification. By given an input image, the task is to generate a vehicle image that has a different pose and keeps the same identity. Thus, it is necessary to make image pairs. Specifically, for each pair of images, the images are of the same vehicle. In the training stage, one image of this pair is input to the generative model and the other image is used for loss calculation. In the test stage, one image is input to the model and the other image is used for performance evaluation. Thus, only the datasets for vehicle re-identification, where each vehicle has several images, are suitable for our task.

1) VeRi-776 : [43], [44] contains many categories with diverse poses. It is a dataset for vehicle re-identification and is collected from real-world surveillance scenarios. It contains over 50,000 images of 776 vehicles and the images are captured by cameras covering 1 km^2 of ground in 24 hours.

2) VehicleID: [50] contains images captured by multiple real-world surveillance cameras distributed in a small city during daytime. In the dataset, there are 26,267 vehicles and 221,763 images in total.

3) Vehicle Keypoint Annotation: We use the public pose labels collected by Wang *et al.* [45] on the VeRi-776 dataset. In this dataset, 20 vehicle keypoints and 8 orientations are annotated. Because this is the only vehicle pose dataset we

TABLE I

QUANTITATIVE METHOD COMPARISONS ON VERI-776 [43], [44] AND VEHICLEID [50]. FOR THE EVALUATION METRICS OTHER THAN FID, A HIGHER SCORE IS BETTER. PERSP. TRANS. REFERS TO ONLY USING THE PERSPECTIVE TRANSFORMATION. CGAN IS THE BASELINE OF THIS PAPER. PAGM IS OUR FULL METHOD, INCLUDING TWO-WAY SKIP CONNECTIONS, PTC AND PERCEPTUAL LOSS. BASELINE+SKIP ONLY APPLIES TWO-WAY SKIP CONNECTIONS. THE DIFFERENCE BETWEEN BASELINE+SKIP AND BASELINE+SKIP+PERCEPTUAL IS THAT BASELINE+SKIP+PERCEPTUAL APPLIES PERCEPTUAL LOSS FUNCTION. BASELINE+SKIP+PTC APPLIES PTC BASED ON TWO-WAY SKIP CONNECTIONS. NOTE THAT WE CANNOT ONLY REMOVE THE TWO-WAY SKIP CONNECTIONS, BECAUSE THEY ARE THE NECESSARY BASIS OF PTC. THE DIFFERENCE BETWEEN PAGM AND BASELINE+SKIP+PTC IS THAT PAGM APPLIES PERCEPTUAL LOSS, WHILE BASELINE+SKIP+PTC DOES NOT

Method	VeRi-776						VehicleID					
	SSIM	IS	mask-SSIM	mask-IS	DS	FID	SSIM	IS	mask-SSIM	mask-IS	DS	FID
Persp. Trans.	0.059	2.087	0.062	2.301	0.285	598.4	0.048	1.031	0.048	1.256	0.357	521.6
CGAN (Baseline) [8]	0.468	2.246	0.615	3.040	0.975	339.2	0.447	1.822	0.622	2.218	**0.997**	325.5
PG2 [38]	0.465	2.451	0.611	2.958	**0.976**	335.7	0.426	1.821	0.611	2.232	0.995	330.4
DSC [40]	0.456	2.521	0.610	2.976	0.970	305.7	0.425	1.864	0.621	2.241	0.995	320.7
Baseline+Skip	0.444	2.374	0.595	3.079	0.968	326.4	**0.455**	1.831	**0.631**	2.261	0.994	315.7
Baseline+Skip+perceptual	0.484	2.386	0.625	2.997	0.960	278.4	0.410	1.900	0.629	2.430	0.993	312.5
Baseline+Skip+PTC	0.474	2.662	0.621	3.611	0.960	316.0	0.410	1.772	0.625	2.358	0.993	322.5
PAGM	**0.492**	**2.662**	**0.630**	**3.612**	0.963	**245.3**	0.444	**1.902**	0.629	**2.431**	0.993	**310.5**
ground truth	1.000	2.852	1.000	3.732	0.929	0.0	1.000	1.992	1.000	2.435	0.982	0.0

could find, we use these annotations to train pose estimation models. We directly use the pose estimation model trained on VeRi-776 to estimate keypoints on VehicleID [50].

4) Evaluation Metrics: The evaluation metrics used in this paper include Structural Similarity (SSIM) [51], Inception Score (IS) [52] and Fréchet Inception Distance (FID) [53]. IS is computed by using a classifier pre-trained on ImageNet [54]. FID improves IS by comparing the statistics of generated images to real ones. Meanwhile, to alleviate the influence of background in evaluation, we use a vehicle mask to calculate mask-IS and mask-SSIM, which have been adopted in [38]. In addition, we use the detection score (DS) generated by the SSD detector [55] to provide another view whether the generated image is realistic or not. For these evaluation metrics other than FID, a higher score indicates that the generated image is more realistic / similar to the ground truth.

B. Implementation Details

1) Vehicle Pose Estimation: Using the annotated vehicle keypoints [45] on VeRi-776, we train a stacked hourglass network (SHN) [15] to predict the 20 vehicle keypoints. The input images are resized to 256×256. We set the batch size to 4 and the learning rate to 0.00025. Weight decay applied to the learning rate is set to 0.96. The other learning settings of SHN follow [15]. We use the manually-annotated points to train the pose estimation model and PAGM. In the testing stage, the pose estimation model generates keypoints for PAGM.

2) Pose Estimation Results: We evaluate the pose estimation performance on VeRi-776 and VehicleID. Following [45], a valid pose detection is defined when it is located within a circle of three pixels from the ground truth position on the final response map. In addition, invisible keypoints are ignored in the evaluation step. Since the VehicleID dataset does not have pose labels, we randomly choose 1,000 vehicle images and carefully label the keypoints as ground truth. On VeRi-776, our pose estimator obtains an 87.1% accuracy. Meanwhile, on VehicleID, the pose estimation accuracy is 80.0%. It indicates that the trained pose estimation model is

TABLE II

ABLATION STUDY ABOUT THREE LOSSES ($\mathcal{L}_{CGAN}$, $\mathcal{L}_{perceptual}$ AND $\mathcal{L}_{L_1}$) ON VERI-776 [43], [44]

Method	SSIM	IS	DS	FID
$\mathcal{L}_{CGAN}$	0.403	2.654	0.968	326.4
$\mathcal{L}_{CGAN} + \mathcal{L}_{L_1}$	0.474	2.662	0.960	316.0
$\mathcal{L}_{CGAN} + \mathcal{L}_{perceptual}$	0.468	2.660	0.960	296.0
$\mathcal{L}_{CGAN} + \mathcal{L}_{L_1} + \mathcal{L}_{perceptual}$	**0.492**	**2.662**	0.963	**245.3**
ground truth	1.000	2.852	0.929	0.0

capable to offer decent vehicle points on the VeRi-776 dataset. Since the pose estimation model is only trained on the VeRi-776 dataset, its performance is worse on VehicleID, mainly due to domain bias.

3) Generator and Discriminator: The images and pose response maps are fixed to 256×256. We adopt the instance normalization [56] after each convolution. For the hyperparameters in Eq. 4, we set $\lambda_1 = 10$ and $\lambda_2 = 0.01$. We use the Adam optimizer [57] with a mini-batch size of 3. We set $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate starts from 0.0002. We train G and D for 100k iterations.

C. Comparison With Related Art

Because vehicle generation has not been extensively studied, in this section, we re-implement and compare with some existing generative models which can be applied to our task. Specifically, we compare our method with perspective transformation, CGAN [8], PG2 [38] and DSC [40] on the VeRi-776 and VehicleID datasets. As DSC [40] use affine transformation and can only shuttle rectangular regions, we reproduce it by cut a vehicle into several rectangles. Similar to PAGM, DSC [40] has $r = 8$ rectangles and each rectangle contains the same 4 keypoints. We also show evaluation outcomes of the ground truths for reference. The comparison is shown in Table II. The qualitative results are shown in Fig. 6. In addition, the testing results with the pose from other cars are shown in Fig. 7.

1) VeRi-776: On the VeRi-776 [43], [44] dataset, the proposed PAGM obtains an SSIM of 0.492, while SSIM of the four competing methods is 0.059, 0.468, 0.465 and
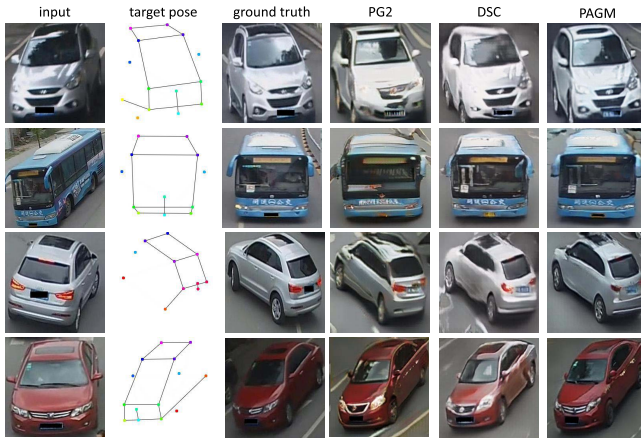
Fig. 6.   Qualitative comparison between our method and previous works: PG2 [38] and DSC [40].

Fig. 7.   Visual results from our model, showing different poses of the same input image. Note that the target pose is extracted from other vehicles.

0.456, respectively. For fair comparison, we also compare mask-SSIM. On this metric, PAGM outperforms the perspective transformation, CGAN [8], PG2 [38] and DSC [40] by +0.568, +0.015, +0.019 and +0.020 in terms of mask-SSIM, respectively. The comparison of SSIM and mask-SSIM indicates that our method can render images that are more similar to the input images. In terms of IS and mask-IS, the proposed method yields an IS of 2.662 and an mask-IS of 3.612 on the testing set. They are higher than results of the competing methods. The FID score of PAGM is 245.3, which is the smallest value among the methods. This indicates that PAGM can generate more realistic vehicle images.

PAGM is superior to DSC, because PAGM can adapt to the characteristics of vehicles. The perspective transformation adopted in PAGM projects quadrilaterals from one perspective to another and can precisely align feature maps. In comparison, DSC [40] can only utilize the affine transformation to shuttle rectangular regions. This is not accurate for vehicles made of irregular quadrilateral planes.

We then evaluate the methods in terms of detection score (DS). The ground truth DS is 0.929. Meanwhile, the DS scores of PG2, CGAN and PAGM are 0.976, 0.975 and 0.960,
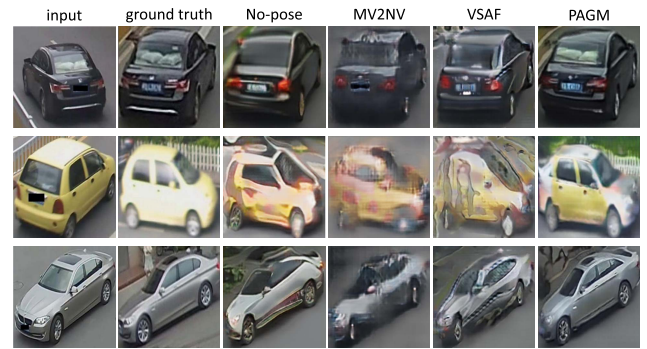

Fig. 8.   Qualitative results of No-pose, MV2NV [13] and VSAF [12] on VeRi-776 [43], [44].

respectively. It clearly indicates that all these methods can produce realistic vehicle images.

The visualizations in Fig. 6 show the effectiveness of PAGM. Because of the introduced perspective transformation in the GAN-based system, our method can transfer the texture details from the input image to the generated image. We observe that our method can generate vehicles that not only *look realistic, but also preserve the original texture details*. We also illustrate some additional visual results from our model, showing the testing results with the pose from other cars.

As shown in Fig. 8, we also compare the proposed method with No-pose, MV2NV [13] and VSAF [12]. In [13], the camera poses are represented by one-hot vectors. In addition, we have also added a method named No-pose. No-pose is similar to the proposed method and uses one-hot encoding representations as the inputs of our method instead of vehicle poses. The results show that our method can generate more realistic vehicle images and have well-preserved details. MV2NV and No-pose can generate a vehicle with a reasonable overall shape but blurry details. VSAF [12] takes viewpoint transformation as input and synthesizes the target view by sampling pixels from source view according to the predicted appearance flow field. The results illustrate that the details generated by our method are more clear and true. The main reason is that our method can accurately transfer local details to the target location.

2) VehicleID: On VehicleID [50], we apply the same model structure used on the VeRi-776 dataset to train on the VehicleID dataset. When we look into the result, we have similar observations to those on VeRi-776. However, the overall results are lower than those in VeRi-776. Since the pose estimator is trained on VeRi-776, pose estimation accuracy is higher on VeRi-776 than VehicleID, which can be explained by the domain gap. We speculate that a better pose estimator would result in a higher quality of the generated images.

D. Component Evaluation

Two components are involved in our method, *i.e.*, the two-way skip connections and PTC. We remove them one at a time to evaluate their contributions respectively. Results are shown in Table II. 1) To evaluate the two-way skip connections, we add them to original baseline model. This method is called "Baseline+Skip". 2) To evaluate PTC, we compare the Baseline+Skip+PTC with Baseline+Skip. Finally,
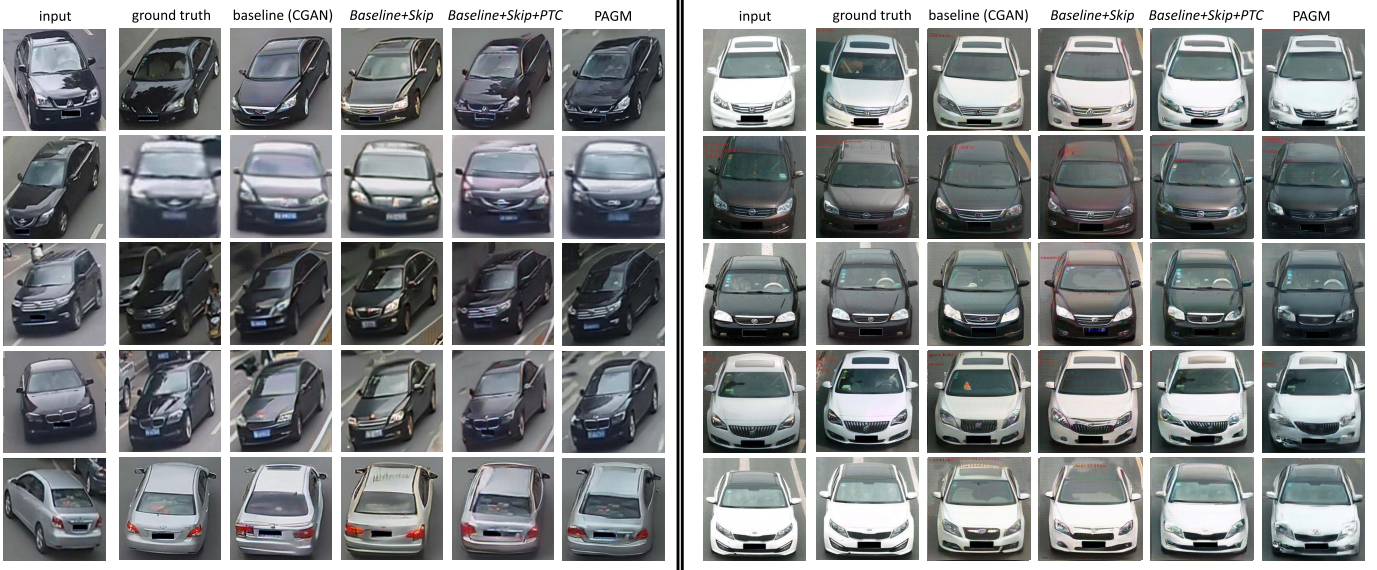
Fig. 9. Sample testing results on VeRi-776 (left) and VehicleID (right). We compare three methods: baseline (CGAN) [8] (Col. 3), Baseline+Skip (Col. 4), Baseline+Skip+PTC (Col. 5) and PAGM (Col. 6). We also show the input image (Col. 1), and ground truth (Col. 2).

3) to evaluate the overall promotion by the two components, we compare Baseline+Skip+PTC with the baseline.

1) The Effectiveness of Two-Way Skip Connections: Two-way skip connections form necessary basis for PTC, so we cannot conduct ablation study for it. Therefore, we first evaluate the two-way skip connections by comparing the baseline (Col. 3) and Baseline+Skip (Col. 4). On VeRi-776, the two-way skip connections bring quantitative improvements in terms of IS and mask-IS. Meanwhile, SSIM and mask-SSIM drop. On VehicleID, the two-way skip connections improve IS and mask-IS by +0.09 and +0.43, respectively. Meanwhile, SSIM and mask-SSIm improve +0.08 and +0.09, respectively. Thus, the quantitative results show that the two-way skip connections can generate better images on the VehicleID dataset.

Second, as shown in Fig. 9, we compare the baseline (Col. 3) and Baseline+Skip (Col. 4) qualitatively. In the images generated by the baseline and Baseline+Skip, the details (e.g, plate, logo, and lamp) are inconsistent with the input images on VeRi-776. However, on the VehicleID dataset, the images generated by Baseline+Skip can preserve some details while images generated by the baseline cannot. The observations on VeRi-776 are different from those on VehicleID. The main reason should be that the poses of the pairs on VehicleID are almost similar (Fig. 9). Thus, the input and target image on VehicleID are to some extent spatially aligned. It is the main reason why only using two-way skip connection is able to preserve some details on the VehicleID dataset.

Another important observation from Fig. 9 is that the color of images generated by Baseline+Skip (Col. 4) is slightly different from the input (Col. 1). The colors of the vehicles seem to be a bit close to the background color. It is mainly because that the two-way skip connections introduce local details as well as the background. The background part leads to a slight bias towards the color of the images generated by Baseline+Skip.

2) The Effectiveness of PTC: We evaluate PTC quantitatively by comparing the results of Baseline+Skip, Baseline+Skip+PTC, Baseline+Skip+Perceptual and PAGM. Note that PAGM is equal to Baseline+Skip+Perceptual+PTC. On VeRi-776, Baseline+Skip+PTC uses PTC and improves SSIM to 0.474 comparing with Baseline+Skip. Meanwhile, IS and mask-IS are improved to 2.662 and 3.611, respectively. Similarly, when using PTC, PAGM obtains an FID of 245.3, while the FID of Baseline+Skip+Perceptual is 278.4. The results indicate that PTC is capable of aligning the local details and generating more realistic images on the VeRi-776 dataset. However, on VehicleID, PTC improves IS and mask-IS while reduces SSIM and mask-SSIM. As mentioned above, the main reason is probably the similar poses of the image pairs on VehicleID.

Then, Col. 4 and Col. 5 of Fig. 9 show qualitative results of Baseline+Skip+PTC and Baseline+Skip. On the one hand, on both VeRi-776 and VehicleID, the details is better preserved in the images generated by Baseline+Skip+PTC. It demonstrates that the aligned feature maps processed by PTC are effective to preserve local details. On the other hand, the color of the image generated by Baseline+Skip+PTC is more realistic and closer to the input image. It demonstrates that, by filtering out the background, PTC is able to reduce color bias.

3) Two Components vs. Baseline: Baseline+Skip+PTC is compared with the baseline to evaluate the promotion by applying two-way skip connections and PTC. Compared with the baseline, IS and mask-IS of Baseline+Skip+PTC improve +0.416 and +0.571, respectively. Meanwhile, SSIM and mask-SSIM of Baseline+Skip+PTC also improve +0.06. As shown in Fig. 9, compared to the baseline (Col. 4), Baseline+Skip+PTC (Col. 6) can render more realistic images and preserve better details. In addition, as shown in Table III, we introduce a vehicle re-identification model to quantitatively evaluate the ability of preserving local textures for

TABLE III

FEATURE DISTANCE RESULTS FOR THE PROPOSED
METHOD AND PREVIOUS METHODS [8], [38], [40]

models	CGAN [8]	PG2 [38]	DSC [40]	PAGM
distance	65.3	56.7	50.9	42.7

TABLE IV

VARIANT STUDY OF PAGM ON THE VERI-776 DATASET.
"ARCHITECTURE" SHOWS THE ARCHITECTURE SKETCH
OF THE VARIANT. BRIEFLY, ONLY THE ENCODER FOR
SOURCE IMAGE AND THE DECODER ARE DEPICTED

Variant	Architecture	mask-SSIM	mask-IS	FID
PAGM-0		0.599	3.079	321.3
PAGM-1		0.619	3.060	309.4
PAGM-2		0.617	3.351	287.4
PAGM-3		0.623	3.541	275.2
PAGM-4		**0.630**	**3.612**	**245.3**
PAGM-5		0.621	3.531	247.2

the components. Following [58], we first train a re-identification model on VeRi-776. We then extract features for the ground truth image and the images generated by CGAN [8], PG2 [38], DSC [40] and our PAGM. Note that the extracted features are 256-dim vectors. Finally, we compute the Euclidean distances between the ground truth and the generated images. A lower score is better. As the results in Table III show that PAGM yields a distance of 42.7, lower than other methods. This indicates that our method can better retain vehicle identities and is advantageous in preserving texture details.

E. Loss Evaluation

We evaluate the effectiveness of the losses in this section. In this paper, as described in Section III-A, we adopt three loss functions: $\mathcal{L}_{CGAN}$, $\mathcal{L}_1$ and $\mathcal{L}_{perceptual}$. As shown in Table IV, only adopting $\mathcal{L}_{CGAN}$ yields the largest FID of 326.4. When introducing $\mathcal{L}_1$, $\mathcal{L}_{CGAN} + \mathcal{L}_{CGAN}$ yields an FID of 316.0. Meanwhile, the FID of $\mathcal{L}_{CGAN} + \mathcal{L}_{perceptual}$ is 296.0. The results indicate that both $\mathcal{L}_{perceptual}$ and $\mathcal{L}_1$ can help improve the generative ability. When combine the three losses together, the system achieves the best performance in terms of FID.

F. Variant Study

Finally, we compare six variants of PAGM, denoted as PAGM-0, PAGM-1, ..., and PAGM-5, on VeRi-776. PAGM-n indicates that n PTCs are applied between the encoder and the decoder. Table IV details the results and the architecture sketch of each variant. Briefly, we use mask-SSIM and mask-IS to evaluate the performance.

First, PAGM-0 has the worst performance compared with other variants. In fact, PAGM-0 has no PTC and is equal to PAGM *w/o* PTC. In our result, PAGM-0 yields a mask-SSIM of 0.599 and a mask-IS of 3.079, which are lower than all the variants that have the PTC module. This suggests that employing PTC is consistently beneficial to vehicle image generation task, regardless of the number of it. It validates our thoughts in Section III-D.

Second, when the number of PTC modules increases from 1 to 4, the result is generally increasing. Mask-SSIM improves from 0.619 to 0.630 and mask-IS from 3.060 to 3.612. However, when the numbers of PTC modules increases from 4 to 5, FID rises from 245.3 to 247.2. It indicates that an appropriate number of PTC modules is important to our system. The reason may be two-fold. On the one hand, when the number of PTC modules is relatively small, the PTC modules are not able to provide sufficiently strong conditions (aligned feature maps). On the other hand, the perspective transformation may perform poorly on small feature maps. In the layer where the fifth perspective transformation module is applied, the map size is 16×16 and perspective transformation cannot generate decent output.

V. CONCLUSION

Given a source vehicle image and a target pose, we present the perspective-aware generative model (PAGM) to deal with the vehicle view synthesis that depicts the same vehicle with target pose. PAGM takes advantages of both conditional generative adversarial nets (CGAN) and perspective transformation. On the one hand, it is based on a CGAN structure that enforces the generated images to look real globally. On the other hand, several perspective transformation modules are applied between the encoder for source image and the decoder. These modules make use of the quadrilateral planes of a vehicle and provide the decoder with aligned feature maps, which are strong constraints to preserve local details. Quantitative and qualitative experiments on the VeRi-776 and VehicleID datasets show that our method can produce realistic images and shuttle carefully the local details from the source to the target.

REFERENCES

[1] Y. Zhou and L. Shao, "Cross-view GAN based vehicle generation for re-identification," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2017, pp. 1–12.

[2] K. Lv *et al.*, "Vehicle reidentification with the location and time stamp," in *Proc. CVPR Workshops*, 2019, pp. 399–406.

[3] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by GAN improve the person re-identification baseline in vitro," 2017, *arXiv:1701.07717*. [Online]. Available: http://arxiv.org/abs/1701.07717

[4] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2107–2116.

[5] W. Deng, L. Zheng, Y. Sun, and J. Jiao, "Rethinking triplet loss for domain adaptation," *IEEE Trans. Circuits Syst. for Video Technol.*, to be published.

[6] Y. Lou, Y. Bai, J. Liu, S. Wang, and L.-Y. Duan, "Embedding adversarial learning for vehicle re-identification," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3794–3807, Aug. 2019.

[7] H. Qiu, F. Ahmad, R. Govindan, M. Gruteser, F. Bai, and G. Kar, "Augmented vehicular reality: Enabling extended vision for future vehicles," in *Proc. 18th Int. Workshop Mobile Comput. Syst. Appl. (HotMobile)*. New York, NY, USA: ACM, 2017, pp. 67–72.

[8] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[9] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[10] K. Rematas, C. H. Nguyen, T. Ritschel, M. Fritz, and T. Tuytelaars, "Novel views of objects from a single image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1576–1590, Aug. 2017.

[11] H. Su, F. Wang, L. Yi, and L. Guibas, "3D-assisted image feature synthesis for novel views of an object," 2014, *arXiv:1412.0003*. [Online]. Available: http://arxiv.org/abs/1412.0003

[12] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 286–301.

[13] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim, "Multi-view to novel view: Synthesizing novel views with self-learned confidence," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 155–171.

[14] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3D view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3500–3509.

[15] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 483–499.

[16] J. Sochor, A. Herout, and J. Havel, "BoxCars: 3D boxes as CNN input for improved fine-grained vehicle recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3006–3015.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016, *arXiv:1611.07004*. [Online]. Available: http://arxiv.org/abs/1611.07004

[19] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[20] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2642–2651.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–16.

[22] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3D models from single images with a convolutional network," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 322–337.

[23] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.* New York, NY, USA: ACM, 2001, pp. 327–340.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[25] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[26] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class Project Stanford CS231N, Convolutional Neural Netw. Vis. Recognit., Winter semester*, vol. 2014, no. 5, p. 2, 2014.

[27] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee, "Weakly-supervised disentangling with recurrent transformations for 3d view synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1099–1107.

[28] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Found. Trends Comput. Graph. Vis.*, vol. 9, nos. 1–2, pp. 1–148, 2015.

[29] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 740–756.

[30] E. Penner and L. Zhang, "Soft 3D reconstruction for view synthesis," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–11, Nov. 2017.

[31] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," 2018, *arXiv:1805.09817*. [Online]. Available: http://arxiv.org/abs/1805.09817

[32] Z. Xu, S. Bi, K. Sunkavalli, S. Hadap, H. Su, and R. Ramamoorthi, "Deep view synthesis from sparse photometric images," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–13, Jul. 2019.

[33] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[34] A. Palazzi, L. Bergamini, S. Calderara, and R. Cucchiara, "Warp and learn: Novel views generation for vehicles and other objects," 2019, *arXiv:1907.10634*. [Online]. Available: http://arxiv.org/abs/1907.10634

[35] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[36] C. Lassner, G. Pons-Moll, and P. V. Gehler, "A generative model of people in clothing," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 853–862.

[37] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, Z. Jie, and J. Feng, "Multi-view image generation from a single-view," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 383–391.

[38] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 406–416.

[39] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 99–108.

[40] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3408–3416.

[41] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3973–3981.

[42] D. Zapletal and A. Herout, "Vehicle re-identification for automatic video traffic surveillance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 25–31.

[43] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 869–884.

[44] X. Liu, W. Liu, H. Ma, and H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.

[45] Z. Wang *et al.*, "Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 379–387.

[46] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, "Learning deep neural networks for vehicle re-ID with Visual-spatio-Temporal path proposals," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1900–1909.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[48] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Munich, Germany: Springer, 2015, pp. 234–241.

[50] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2167–2175.

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[52] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[53] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[54] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[55] W. Liu *et al.*, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37.

[56] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*. [Online]. Available: http://arxiv.org/abs/1607.08022

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[58] B. He, J. Li, Y. Zhao, and Y. Tian, "Part-regularized near-duplicate vehicle re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3997–4005.

Zhang Xiong received the B.S. degree from Harbin Engineering University in 1982 and the M.S. degree from Beihang University, China, in 1985. He is currently a Professor and a Ph.D. Supervisor with the School of Computer Science and Engineering, Beihang University. He is also working on computer vision, information security, and data vitalization.

Kai Lv received the B.S. degree from the School of Computer Science and Technology, Tianjin University of Science and Technology, Tianjin, China, in 2012. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Beihang University, Beijing, China.

Wei Li (Member, IEEE) graduated from the Department of Mathematics and Mechanics, Peking University, in 1966. He received the Ph.D. degree from the University of Edinburgh in 1983. He has been teaching at Beihang University (formerly Beijing Institute of Aeronautics) since 1966. He has served as the President of Beihang University from 2002 to 2009. He was elected as a member of the Chinese Academy of Sciences in 1997. He has been a Professor with the School of Computer Science and Engineering, Beihang University, since 1986. He is currently a computer scientist. He currently serves as the Director of the State Key Laboratory of Software Development Environment and a member of the National Educational Advisory Committee.

Hao Sheng (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Computer Science and Engineering, Beihang University, China, in 2003 and 2009, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University. He is also working on computer vision, pattern recognition, and machine learning.

Liang Zheng (Member, IEEE) received the B.E. degree in life science and the Ph.D. degree in electronic engineering from Tsinghua University, China, in 2010 and 2015, respectively. He was a Post-Doctoral Researcher with Centre for Artificial Intelligence, University of Technology Sydney, Australia. He is currently a Lecturer and a Computer Science Futures Fellow with the Research School of Computer Science, Australian National University. His research interests include image retrieval, classification, and person re-identification.