Photorealism in Driving Simulations: Blending Generative Adversarial Image Synthesis with Rendering

Ekim Yurtsever, *Member* Dongfang Yang, *Student Member* Ibrahim Mert Koc, *Student Member* and Keith A. Redmill, *Senior Member*

***Abstract*—Driving simulators play a large role in developing and testing new intelligent vehicle systems. The visual fidelity of the simulation is critical for building vision-based algorithms and conducting human driver experiments. Low visual fidelity breaks immersion for human-in-the-loop driving experiments. Conventional computer graphics pipelines use detailed 3D models, meshes, textures, and rendering engines to generate 2D images from 3D scenes. These processes are labor-intensive, and they do not generate photorealistic imagery. Here we introduce a hybrid generative neural graphics pipeline for improving the visual fidelity of driving simulations. Given a 3D scene, we partially-render only important objects of interest, such as vehicles, and use generative adversarial processes to synthesize the background and the rest of the image. To this end, we propose a novel image formation strategy to form 2D semantic images from 3D scenery consisting of simple object models without textures. These semantic images are then converted into photorealistic RGB images with a state-of-the-art Generative Adversarial Network (GAN) which is trained on real-world driving scenes. This replaces repetitiveness with randomly generated but photorealistic surfaces. Finally, the partially-rendered and GAN synthesized images are blended with a blending GAN. We show that the photorealism of images generated with the proposed method is more similar to real-world driving datasets such as Cityscapes and KITTI than conventional approaches. This comparison is made using semantic retention analysis and Frechet Inception Distance (FID) measurements.**

***Index Terms*—Driving simulation, deep learning, generative adversarial networks, image synthesis**

I. Introduction

DRIVING simulations are important for developing and evaluating intelligent transportation systems [1]. A good simulation environment should have accurate vehicle dynamics, realistic traffic behavior, and high visual fidelity. Visual fidelity is especially important for validating vision-based algorithms and conducting human-in-the-loop experiments. There are numerous studies [2], [3], [4], [5], [6], [7] that utilize a driving simulation whose integrity greatly depends on the visual quality of the simulation environment.

The aforementioned studies all use rendered images that are generated by a simulation environment. However, limited works has been done on evaluating and improving the visual fidelity of the state-of-the-art driving simulators. Here we investigate a new approach: introducing generative photorealism to virtual driving environments using deep learning. Data-centric applications trained or fine-tuned in a photorealistic driving simulation can be more confidently deployed to the real world. Furthermore, automated driving systems can be tested with photorealistic-looking dangerous scenes that are difficult to obtain outside a simulation environment. In addition, if non-realistic repetitive patterns can be replaced by photorealistic scenery, the degree of immersion for human-in-the-loop simulation experiments can be increased.

Authors are with the Department of Electrical and Computer Engineering, the Ohio State University, Columbus, OH, 43210 USA (email: yurtsever.2@osu.edu).

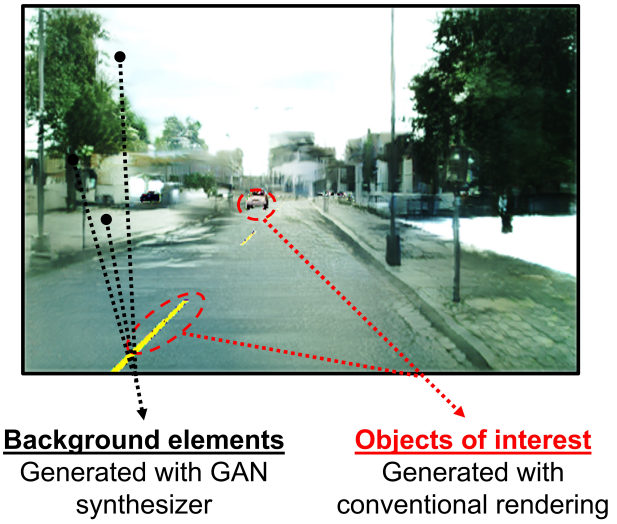


Fig. 1: The proposed framework generates photorealistic imagery for driving simulators. First, we obtain the semantic layout of the scene through a conventional simulation pipeline with textureless simple 3D models. Then, this semantic layout is converted into a photorealistic RGB image using GANs with the proposed image formation and blending strategy.

The fidelity of a conventional driving simulator depends on the quality of its computer graphics pipeline, which consists of 3D models, textures, and a rendering engine. High-quality 3D models and textures require artisanship, whereas the rendering engine must run complicated physics calculations for the realistic representation of lighting and shading [8]. These processes are labor-intensive, and images obtained this way are not photorealistic. Here we investigate alternatives for alleviating the aforementioned costs. An overview of our approach is shown in Figure 1.

The alternative to rendering is neural network based generative adversarial image synthesis. The advent of Generative

Adversarial Networks (GAN) [9] enables realization of photo-realistic image synthesis [10], [11], [12], [13], [14], [15], [16]. A particular sub-problem, conditional image synthesis [17], [18], [19], [20], [21], delves into the more specific task of mapping a pixel-wise semantic layout to a complying photo-realistic image. The conditional semantic layout is the key link between the 3D scene and the generative synthesizer in our framework. More recently, video-to-video synthesis [22] was proposed as an alternative to image synthesis. The temporal dimension was added to the generative process to reduce inconsistencies between synthesized frames.

The main motivation of this work is twofold: to increase the visual fidelity of driving simulations and reduce the manual labor requirements for 3D mesh and texture creation. With the use of GAN-based photorealistic image synthesizers, background objects such as trees, mountains, and the sky can be generated without detailed meshes or texture information. However, conventional rendering is still needed to have full control over important objects of interest, such as vehicles and road markers.

In this paper, we propose to integrate generative adversarial image synthesis into a driving simulation. For each time step, CARLA, an open-source driving simulator [23], determines the scene's semantic layout with simple, textureless 3D models that are radiant with a unique class color. It should be noted that there is no illumination source other than the radiant 3D objects and no reflections or ambient occlusion are considered at this step. Then, a virtual pinhole camera is used to form a 2D semantic image from this scene. This image is the equivalent of a pixel-wise semantic segmentation mask. Next, the GAN-based image synthesizer converts the 2D semantic image to a photorealistic image. Conditional GAN (cGAN) [24] and CYcle GAN (Cy-GAN) [17] are the main techniques for this step. Simultaneously, a few objects of interest are partially rendered using a conventional rendering engine [25]. This is necessary as full control over some critical objects, such as lane markings and vehicles in a driving scene, is only achieved with a conventional graphics pipe. Finally, a blending GAN mixes the cGAN/Cy-GAN synthesized image with the individually rendered objects. The proposed method was evaluated with semantic segmentation [26], an important driving-related perception task.

The main contributions of this work are:

- A novel driving simulation graphics pipeline for expediting scene creation using automated synthesis of background elements such as buildings, vegetation, and sky.
- Replacing recurring patterns, such as repeating tree and building models, that are common in driving simulations with generative photorealistic surfaces as shown in Figure 2. Repetitive patterns can break immersion for human-in-the-loop simulation experiments. In addition, machine learning algorithms trained or fine-tuned in a repetitive environment can fail in the real world due to overfitting. As such, the proposed approach aims at increasing the integrity of simulation-based intelligent transportation research.
- Blending generative adversarial image synthesis with physics-based partial rendering.

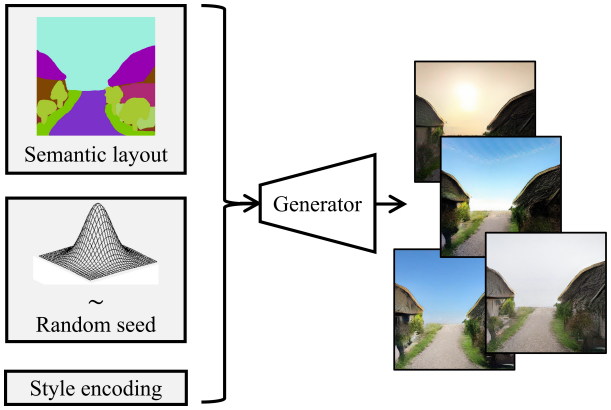


Fig. 2: We first create the semantic layout, and then use SPADE [19] with different style encodings to generate random but photorealistic RGB background imagery. Repetitive patterns that are common in driving simulations are memorizible by learning algorithms and break immersion for human driver subjects. The proposed approach alleviates these shortcomings.

II. Related Work

Simulation based driving studies. Human driver reaction to various driving-related stimuli has been observed via simulation environments in numerous studies. The simulation's visual fidelity is critical for such experiments, as humans are accustomed to a real-world driving setting. Driving simulators have been used to study the driver's reaction during an automated driving take-over [4], to monitor human responses to stressful driving stimuli [2], to find the effect of inter-vehicular distances on human car following behavior [5], and to measure the effect of acoustic cues on situational awareness of human drivers [7]. An automated highway driving system with human-like decision-making capabilities have been developed via a driving simulator [6]. Another study [3] focused on human pose estimation using simulated images and showed that data-centric algorithms fine-tuned in these simulations could be used in real-world scenarios.

A recent study showed that human subjects gaze with higher variance and exhibit more diverse steering activity in driving simulations that have better visual fidelity [27]. Higher visual fidelity is always desired in human-in-the-loop experiments because human driving behavior deviates from real-world behavior in unrealistic simulation environments. [28]. Furthermore, data-centric methods that are trained on synthetic data generated by conventional rendering engines fail to perform with real-world images [29].

The number and significance of these studies underline the importance for improving the visual fidelity in driving simulations. New technologies can be developed more effectively with a better simulator. For example, if photorealism can be achieved, a learning-based lane-boundary detection algorithm [30] can be trained in a simulation and deployed in the real world.

Rendering. Physics-based rendering [8] has been used at the end of the line of conventional computer graphics pipelines

to form 2D imagery from virtual 3D scenes for a long time. The most common approaches, rasterization and ray-tracing, require a full pipeline of detailed 3D models, their surface textures and materials, and a physics engine such as Unreal Engine 4 [25] to run complicated calculations for representing light and shading. Here, we propose to partially replace this pipe with much simpler 3D models and remove light, texture, and material information for most of the objects in the scene. We also show that the visual fidelity can be increased with the proposed method.

Neural rendering. Recent work [31] demonstrated that 2D image formation could be achieved given a camera pose and light position in a 3D scene using differentiable convolutional networks. The key enabler here is the formulation of the discrete rasterization problem [32]. With a differentiable rendering framework, a neural network can be trained with backpropagation. There is additional work [33], [34], [35] focusing on the different aspects of differentiable rendering formulation and approximations. Neural rendering is a promising technique. However, this approach still requires detailed 3D models and is incapable of generating texture information, which reduces the visual fidelity of the output. In comparison, we propose to use generative models for reducing 3D model and texture complexity.

Generative adversarial image synthesis. Generative adversarial image synthesis omits rasterization and rendering. Physical phenomena such as lighting and reflectivity are completely ignored by GAN based neural image synthesizers [10], [11], [12], [13], [14], [15], [16]. Instead, the photorealism is achieved by training the GAN with real-world data. In other words, the network learns to generate photorealistic images by capturing a latent probability distribution underlying real-world datasets. This approach has one major drawback: there is no constraint on the semantic layout of the generated 2D image. Hence, no association with 3D scenery can be constructed. As such, this methodology cannot be applied for our image formation purposes.

Conditional generative adversarial image synthesis. On the other hand, conditional GANs [17], [18], [19], [20], [22], [21], [36] have been effectively used for image synthesis while retaining a semantic constraint. Typically, this constraint is a pixel-wise semantic segmentation mask, but other modalities such as text [37] have also been used. One limiting factor for Conditional GAN (cGAN) is the paired data requirement. The dataset must contain semantic segmentation masks and the corresponding real-world images. Building such paired datasets is labor-intensive because every real-world image needs a corresponding semantic segmentation label assigned by a human annotator.

Cycle-consistency and domain adaptation. Cycle consistent GANs and unsupervised domain adaptation techniques make the paired dataset requirement uneccessary [38], [29], [39], [40], [41], [42], [39]. These works have illustrated that high fidelity image synthesis can be achieved with unpaired data also. Cycle-consistency is very promising and has a huge application range. For example, CyCADA [29] can translate an existing game engine generated image into a photorealistic image.

The aforementioned GAN-based image synthesis techniques have not been integrated into driving simulation pipelines until now. This contribution makes our proposed method novel. We propose to use simple 3D models radiant with unique class color-codes without textures to form a 2D semantic image. This image is analogous to a 2D semantic segmentation mask. Then, a state-of-the-art GAN-based image synthesizer trained on real-world datasets is used to generate RGB imagery. We tried both cGAN and Cy-GAN variants. Additionally, we render certain important objects of interest, such as cars in an urban scene, with Unreal Engine 4. Images obtained by blending the partial-render foreground and GAN background are more realistic. Blended images also retain the semantic layout of the scene better.

III. Preliminaries

Generative Adversarial Networks (GAN) [9] use a generator G and a discriminator D in a simultaneous adversarial training strategy. The goal of G is to generate data $\hat{x}$ that is indistinguishable from the real data $x \in X$. During training, G captures the probability distribution p_{data} which should closely match the distribution underlying the real data. This is achieved by training a generative mapping function $G(z)$ that maps an a priori noise distribution $p_z(z)$ to the data domain X. While G tries to generate the most realistic $\hat{x}$, the discriminator D tries to discriminate fake data $\hat{x}$ from real data x. The output of $D(x)$ is the probability that x is real. $G(z)$ and $D(x)$, both of which are neural networks, are trained simultaneously with the following min max function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[\log D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}\left[\log(1-D(G(\boldsymbol{z})))\right]. \quad (1)$$

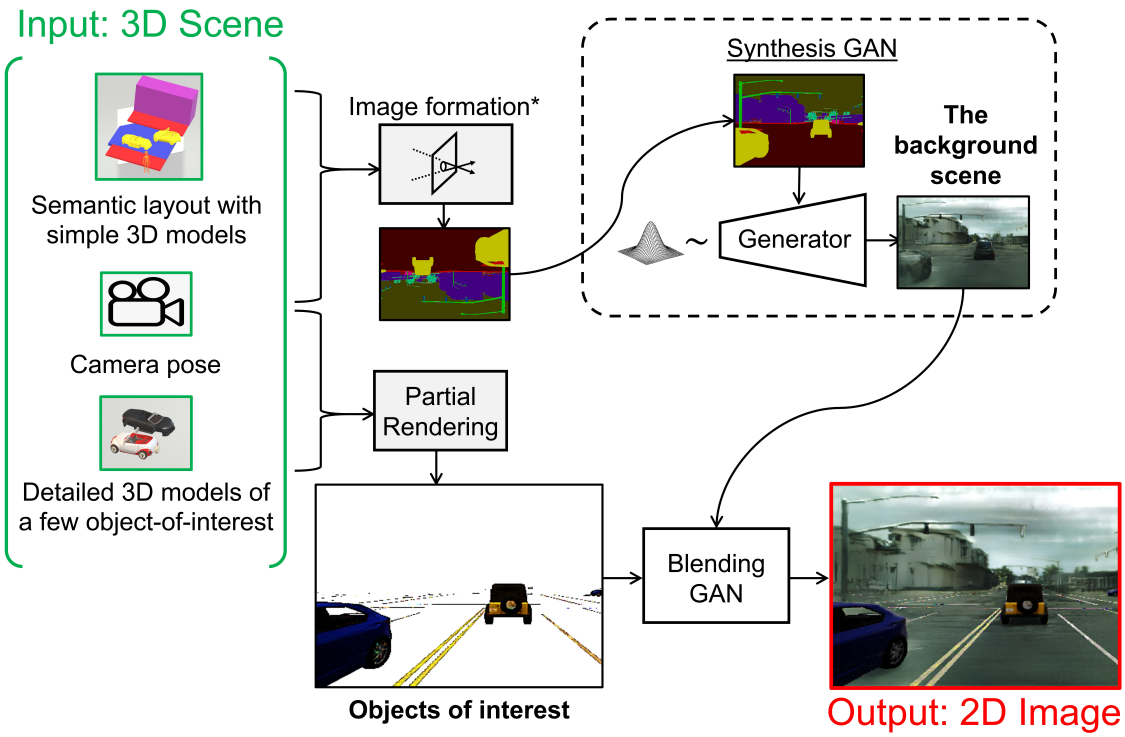
IV. Method

A. Problem formulation

We define a virtual 3D driving scene S with a 6-tuple $(O_1, O_2, P_1, P_2, T_2, \mathbf{x})$. Where $O_1 = (\mathbf{o}_{1,1}, \mathbf{o}_{1,2}, \cdots, \mathbf{o}_{1,n})$ is a list of object pose vectors, $\mathbf{o} \in \mathbb{R}^6$, and $P_1 = (M_1, M_2, \cdots, M_n)$ is the list of corresponding simple object meshes. We assume P_1 is radiant with unique class color-codes. O_2 is a sublist of O_1 for certain objects of interest, and it has a corresponding list of more complicated object meshes P_2. P_2 is not radiant. T_2 is a list of texture maps that corresponds to P_2. $\mathbf{x} \in \mathbb{R}^6$ is the pose vector of a virtual camera. It should be noted that a corresponding T_1 to O_1 does not exist.

We follow the formal definition of a triangular mesh given in [43]. $M := (V, Q)$ is a triangular mesh defined with faces $Q \subseteq \{1, \cdots, |V|\}^3$ and vertices $V \subseteq \mathbb{R}^3$. Where $q = (q_1, q_2, q_3) \in Q$ is a triangular face with corresponding vertices v_{q_1}, v_{q_2}, and v_{q_3}. $E(Q)$, edges between the vertices are defined by faces implicitly.

Problem 1. Given S, we are interested in finding a mapping function $U : \mathbf{x} \rightarrow \mathbb{R}^{H\times W\times 3}$ that will convert the camera pose vector $\mathbf{x}$ to a photo-realistic RGB image with height H and width W.



*all objects are opaque and radiant with unique class color in the 3D semantic layout scene. No ambient occlusion is considered. Then, a pinhole camera can easily convert the 3D scene into a corresponding upside-down semantic image.

Fig. 3: Overview of the proposed method. We introduce a novel neural graphics pipeline to form 2D image representations from virtual 3D scenes. Most of the scene is generated with very simple 3D models without texture except for a few partially rendered objects of interest. We then blend the cGAN synthesized image with a physics-based partial render for increasing visual fidelity *and* to maintain full control over the appearance of objects of interest.

The overview of our solution, Hybrid Generative Neural Graphics (HGNG) is shown in Figure 3 and Algorithm 1, and the formal description follows.

Algorithm 1: HGNG($O_1, O_2, P_1, P_2, T_2, \mathbf{x}$)

Input:
O_1, the list of object pose vectors
P_1, the list of simple object meshes w/o texture.
O_2, a sublist of O_1, corresponds to objects of interest
P_2, the list of complex object meshes.
T_2, the list of texture maps that corresponds to P_2, O_2.
$\mathbf{x}$, the pose vector of the pinhole camera
Output:
$\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, 2D RGB image.
Main algorithm:
$\mathbf{m} = h_{\text{pinhole}}(O_1, P_1, \mathbf{x})$;
$\mathbf{I}_{\text{background}} = f_{\text{generator}}(\mathbf{m}, z \sim N)$;
foreach $i(1, 2...n)$ **do**
  $\mathbf{I}_i^{\text{object-of-interest}} = L_{\text{rendering}}(O_2(i), P_2(i), T_2(i))$;
end
$\mathbf{I}_{\text{foreground}} = \sum_i^n \mathbf{I}_i^{\text{object-of-interest}}$;
$\mathbf{I} = b_{\text{generator-blending}}(\mathbf{I}_{\text{background}}, \mathbf{I}_{\text{foreground}})$;

B. Semantic Image formation

A semantic image formation function h can be obtained with O_1, P_1 and a pinhole camera model. Let $\mathbf{m} \in \mathbb{M}^{H \times W}$ be a pixel-wise semantic image whose entries correspond to the semantic classes of the scene. Then $h : \mathbf{x} \rightarrow \mathbb{M}^{H \times W}$ maps $\mathbf{x}$ to an integer subspace ($\mathbb{M} \subset \mathbb{Z}$) using the pinhole camera model [44] given by:

$$\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} = -\frac{d}{p_3} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \tag{2}$$

where (p_1, p_2, p_3) are the 3D coordinates of point $\mathbf{p}$ in $\mathbb{R}^3$, (m_1, m_2) are the corresponding pixel coordinates in $\mathbf{m}$, and d is the distance between the focal point and image formation plane. $\mathbf{m}$ is an upside-down image as shown in Figure 3. $\mathbf{m}$ is rotated 180° for the next step. For simplicity, we use the same notation $\mathbf{m}$ for the rotated image in the remainder of the paper.

Then, the problem narrows down to finding $f : \mathbf{m} \rightarrow \mathbb{R}^{H \times W \times 3}$. This is the exact same goal of the well-studied [17], [18], [19], [20] conditional image synthesis problem.

C. Generative Adversarial Image Synthesis with cGANs and Cy-GANs

We propose to use the generator networks of cGANs or Cy-GANs to map $G : \mathbf{m} \rightarrow \mathbb{R}^{H \times W \times 3}$. Training is to be done on

a real-world paired dataset (RGB image, pixel-wise semantic mask) for cGAN, while Cy-GANs can be trained with an unpaired dataset (RGB images), (pixel-wise semantic masks).

cGAN [24] extends the original GAN and can generate realistic fake data while retaining a conditional constraint. This is achieved by pairing the conditional constraint y with the data x and creating a new paired dataset (x, y). This pair can be an (RGB image, pixel-wise semantic layout image), and (image, text), and so on. x and y do not have to share the same modality. Details of cGAN can be found in [24]. cGAN can successfully generate photo-realistic fake data with a conditional constraint. However, the paired dataset requirement increases the cost of this approach.

In comparison, building an unpaired X and Y is relatively easy. Cycle GAN (Cy-GAN) [41] enables photo-realistic image synthesis with unpaired data. In summary, Cy-GAN contains two generators, $G(x)$ and $F(y)$, which map $X \rightarrow Y$ and $Y \rightarrow X$ respectively. Also, two discriminators, D_X and D_Y, try to distinguish fake data from real data. The adversarial losses are similar to the original GAN; the addition is the novel cycle consistency loss. This loss prevents the mappings of G and F from diverging from each other. The key idea of cycle GAN is the use of two generators to create a cycle. First, $G(x)$ generates fake $\hat{y}$, then $F(G(x))$ translates the fake $\hat{y}$ back to $\hat{x}$. If the cycle is consistent, then $x \approx \hat{x}$.

The baseline cGAN employed in this study is a SPatially-Adaptive-(DE)-normalization (SPADE) [19] network, which is a state-of-the-art cGAN based image synthesizer. SPADE outperforms other image-to-image synthesizers by retaining semantic information against conventional normalization operations [19]. This is achieved through the following de-normalization operation where the activation value at layer i is given by:

$$\gamma^i_{c,y,x}(\mathbf{m})\frac{h^i_{n,c,y,x} - \mu^i_c}{\sigma^i_c} + \beta^i_{c,y,x}(\mathbf{m}) \tag{3}$$

where $h^i_{n,c,y,x}$ is the activation before normalization and μ^i_c and σ^i_c are the mean and standard deviation in channel c. $\gamma^i_{c,y,x}(\mathbf{m})$ and $\beta^i_{c,y,x}(\mathbf{m})$ are learned variables that modulates the normalization process. We refer the readers of the original SPADE paper [19] for more details.

We use a pre-trained SPADE on the Cityscapes dataset [45] as the mapping function f_s and obtain the synthesized image with it $\mathbf{I} = f_s(\mathbf{m})$.

D. Partial rendering

To increase visual fidelity and have full control over certain objects of interest, we propose using physics-based rendering to obtain partially-rendered images $\mathbf{I}_r$. Besides O_2, P_2, T_2 and $\mathbf{x}$, a light source is also needed for rendering. Here we assume that the properties and location of the light source are fixed and known relative to $\mathbf{x}$. Then the rendering equation [8] can be used to render objects of interest.

$$L_0(\mathbf{p}, \omega, \lambda, t) = L_e(\mathbf{p}, \omega_0, \lambda, t) + \int_\Omega f_r(\mathbf{p}, \omega_i, \omega_0\lambda, t) L_i(\mathbf{p}, \omega_i, \lambda, t)(\omega_i.\mathbf{n}) d\omega_i \tag{4}$$

where $L_0(\mathbf{p}, \omega, \lambda, t)$ is the total spectral radiance, λ is wavelength, ω_0 is the outgoing light direction, ω_i is the incoming light direction, t is time and $\mathbf{p}$ is a point in 3D space. $L_e(\mathbf{p}, \omega_0, \lambda, t)$ is the emitted spectral radiance, Ω is a unit hemisphere with the surface normal center $\mathbf{n}$ of $\mathbf{p}$ and it contains all values for ω_i, $f_r(\mathbf{p}, \omega_i, \omega_0\lambda, t)$ is the bidirectional reflectance function and finally $L_i(\mathbf{p}, \omega_i, \lambda, t)$ is the spectral radiance of the incoming wavelength.

With Equation 4, the spectral radiance of each 3D point on a few objects of interest is obtained. Then, the partially rendered image $\mathbf{I}_r$ is formed with the same pinhole camera model introduced in Equation 2.

E. Blending

Here we propose to blend the synthesized image $\mathbf{I}$ with the partially rendered image $\mathbf{I}_r$ to obtain a hybrid image $\mathbf{I}_h$ as shown in Figure 4. The hybrid image is defined as:

$$\mathbf{I}_h := b(\mathbf{I}, \mathbf{I}_r). \tag{5}$$

where the blending function $b : (\mathbf{I}, \mathbf{I}_r) \rightarrow \mathbb{R}^{HxW\times 3}$ maps the synthesized and partially rendered images to a new hybrid RGB image. We compared three different blending functions b in this study.

Alpha blending. Taking $\mathbf{I}$ as the background image and $\mathbf{I}_r$ as the foreground image, the alpha blended image $\mathbf{I}_h$ can be obtained with:

$$\mathbf{I}_h = \alpha\mathbf{I} + (1 - \alpha)\mathbf{I}_r. \tag{6}$$

Pyramid blending. With the gaussian pyramid mask G_R [46], L_a the laplacian pyramid of the foreground $\mathbf{I}_r$, and L_b laplacian pyramid of background $\mathbf{I}$, the laplacian blended pixel $b(i, j)$ can be obtained with:

$$b(i, j) = G_R(i, j)L_a(i, j) + (1 - G_R(i, j))L_b(i, j). \tag{7}$$

GAN blending. As a third blending option, we employed GP-GAN [47]. The generator of GP-GAN converts a naive copy-paste blended image to a realistic well-blended image. Besides conditional GAN loss, GP-GAN employs an auxiliary l_2 loss to sharpen the image.

$$\mathcal{L}(x, x_g) = \lambda\mathcal{L}_{l_2}(x, x_g) + (1 - \lambda)\mathcal{L}_{\text{adv}}(x, x_g) \tag{8}$$

where $\mathcal{L}(x, x_g)$ is the final loss, $\mathcal{L}_{l_2}$ is the l_2 loss and $\mathcal{L}_{\text{adv}}$ is the adverserial loss. λ is a hyperparameter and set to 0.999.

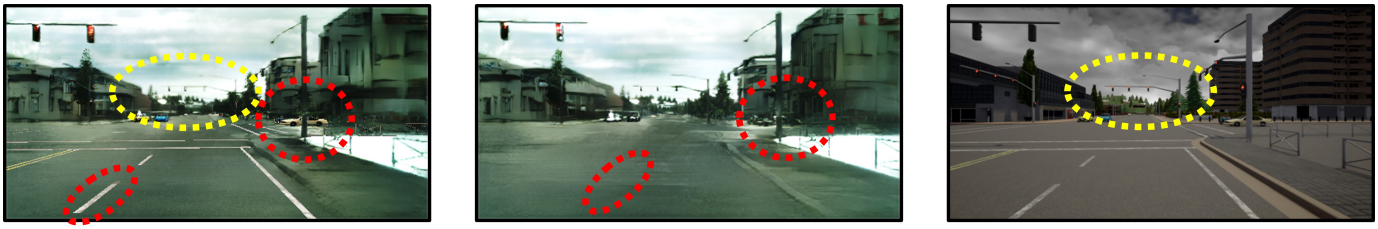


Fig. 4: The proposed framework (a) converts the semantic layout of the scene into a photorealistic image by blending partially rendered foreground objects with a GAN generated background. The conventional rendering engine [23] (c) requires detailed models and texture information while outputting unrealistic background trees and vegetation (shown with a yellow circle). On the other hand, using only a cGAN (b) [19] approach leads to poor car shapes and omitting road markings (shown with a red circle), while removing the need for texturing and rendering calculations. The proposed method (a) has the best of both worlds.

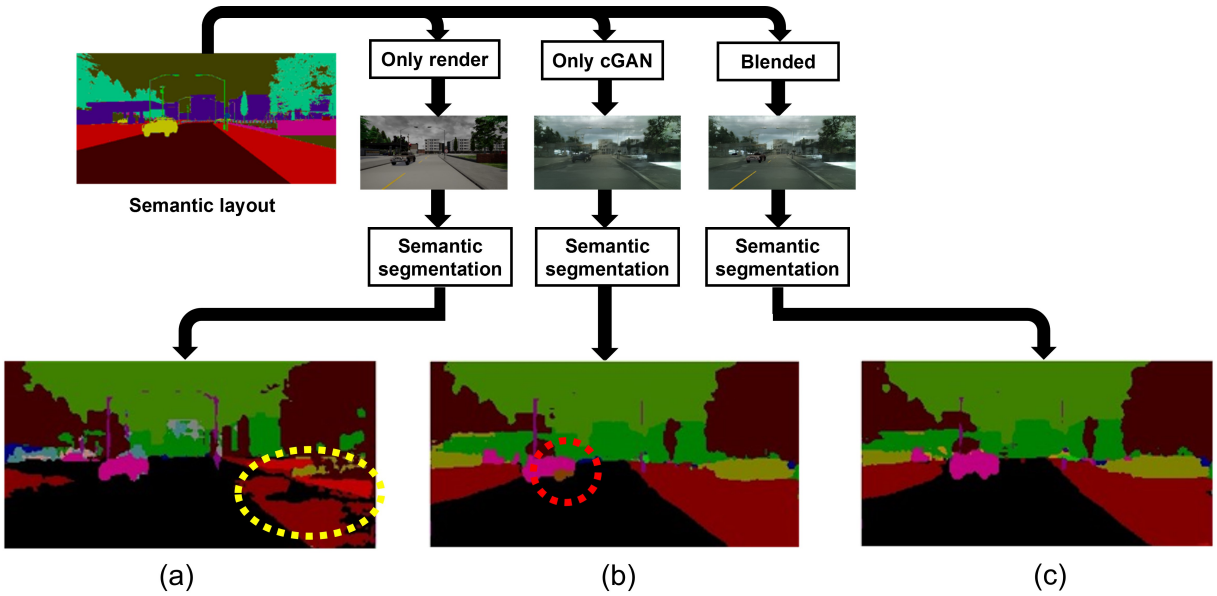


Fig. 5: An illustration of semantic retention analysis. The semantic segmentation result should stay true to the initial semantic layout. (a) Full-render yields unrealistic shadows. On the bottom right-hand side of the left-most image (shown with a yellow circle), shadows of trees cast on the sidewalk were misclassified as a road by DeepLabV3. (b) cGAN generated vehicles do not retain their shapes perfectly (middle image, shown with a red circle). (c) Blending retains the semantic relationship with the source layout (right-most image). This figure employs different color codes to distinguish the semantic layout formation and semantic segmentation processes for illustration purposes.

V. Experiments

A. Implementation details

We used the SPADE implementation provided by the original authors [19]. The network was trained on Cityscapes [45], an urban driving dataset with paired semantic mask and image data. CARLA [23], an open-source driving simulator built upon Unreal Engine 4 was utilized to obtain the semantic layout and partially rendered images. We used the shading and lighting engine [25] of Unreal Engine 4 in our experiments. Only vehicles and lane markings were considered as objects of interest. For blending, we used a GP-GAN [47] trained on the Transient Attributes Database [48]. All computational experiments were conducted with an Nvidia RTX 2080.

B. Evaluation

1) Semantic retention: Figure 5 illustrates the semantic retention analysis, a common [18], [21], [19] evaluation method for fake image synthesis. Semantic retention measures the semantic correspondence between the conditional semantic mask and the synthesized image. In summary, an external semantic segmentation network is used to segment the synthesized image. Then, the discrepancy between the conditional semantic layout (input of the synthesizer) and the semantic mask obtained from the generated image (output of the pre-trained external segmentation network) is calculated with top-1 accuracy. A good synthesizer should produce photo-realistic images while retaining the initial conditional semantic layout. In other words, the initial semantic layout is accepted as the ground truth, and

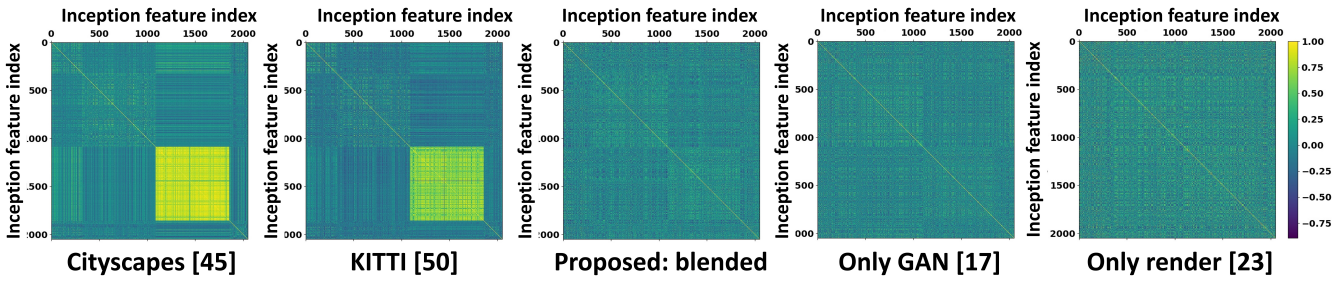


Fig. 6: InceptionV3 feature vector correlation matrices of real and synthetic data. The synthetic dataset that was generated with the proposed blending approach shows a similar correlation pattern with real data. This pattern does not emerge with the only render or only GAN methods.

the image synthesizer's mask accuracy is calculated to obtain the retention score. A higher retention score is favorable.

In this study, we employed DeepLabV3 [26], a state-of-the-art semantic segmentation network, to measure semantic retention. The network was trained on Cityscapes, an urban driving dataset [45].

2) FID: Frechet Inception Distance (FID) [49] is a commonly used [19], [22] performance metric for measuring visual fidelity. In summary, a deep neural network is employed to extract features of all images in a dataset. Then, the covariance and mean of features obtained from synthesized and real datasets are compared to generate a score. We do not have any real-data corresponding to our virtual 3D scene, but FID can still be used with unpaired data. As such, three different real-world datasets [45], [50], [51] were utilized as the ground truth.

An InceptionV3 [52] model that was trained on ImageNet [53] was employed as the feature extractor. After features were extracted from the synthesized images and real-world images from Cityscapes [45], KITTI [50], and ADE20K [51], the FID is calculated as follows:

$$d^2 = ||\mu_1 - \mu_2||^2 + Tr(C_1 + C_2 - 2\sqrt{(C_1 C_2)}) \quad (9)$$

where μ_1, μ_2 are the means of features, and C_1, C_2 are the covariances obtained from datasets 1 and 2 respectively, where the first dataset consists of real images and the second synthesized images. The smaller the distance d^2, the more similar are the two datasets. In other words, a small FID indicates that fake data is similar to real-world data.

The synthesized images were then compared against each other using FID scores as shown in Table II. μ_1 and C_1 were obtained from the real datasets and do not change in a column, whereas μ_2 and C_2 were obtained from synthesized images and vary with each row. A lower FID indicates high visual fidelity.

3) Inception score: Inception Score (IS) was initially proposed to evaluate the generator performance of GANs [54]. In summary, a pre-trained image classifier is run over a GAN generated fake dataset. The distribution of predicted classes, along with the confidence intervals, were then compared against a real dataset. A higher Inception Score indicates higher image quality and diversity. IS differs from FID with the use of actual classification results, whereas FID utilizes latent features. Details of IS can be found in [54].

4) Comparisions and ablations: Ablation studies were conducted to demonstrate the effect of each component of the proposed method. The ablation list is:

1) No partial-render (only vanilla cGAN or Cy-GAN)
2) No cGAN or Cy-GAN (only full render)
3) Alpha blend (cGAN or Cy-GAN + partial render)
4) Pyramid blend (cGAN or Cy-GAN + partial render)
5) GAN blend (cGAN or Cy-GAN + partial render)

We also compared the performance of 2 alternative generative neural image synthesizers: SPADE [19] and Cycle-GAN (Cy-GAN)[17].

We used vanilla CARLA [23] to obtain fully rendered images of an urban scene. The semantic layout of the scene was also imported from CARLA and used as the conditional input for the generative adversarial image synthesizers. Only vehicles and lane markings were considered by the partial-renders.

In this work, the image synthesis was done frame-by-frame.

C. Results

1) Qualitative results: The qualitative results are shown in Figures 4, 5, and 6. These figures illustrate fully rendered, blended, and only cGAN images. As can be seen in Figure 5, rendered shadows are unrealistic, while only cGAN generated vehicles cannot retain their shapes. These results underline the importance of partial rendering of objects of interest such as cars, vans, and lane markings. The hybrid approach combines the accuracy of a full-render with the realism of a generative model.

Treating foreground objects differently from background scenery with the proposed blending technique improves the photorealism of the final image as can be seen in Figures 4, 5, 6. The appearance of foreground objects needs to be controllable and rendered in detail. This necessitates conventional rendering with high-detail models and textures. However, the background scenery does not need to be controllable at the same level of detail. Hence, GAN-based image synthesizers can be used to automate background scene generation. Our use of a GAN-based image synthesizer completely removes the texture and detailed model requirements of background scenery generation while increasing visual fidelity. In Figure 4, the conventional

TABLE I: Semantic retention performance- higher scores are better. Our methods outperform the physics-based rendering approach.

Method	Semantic retention
Baseline: only render [23]	0.819
only Cycle GAN [17]	0.343
Proposed	
cy-GAN alpha blend	0.362
cy-GAN pyramid blend	0.353
cy-GAN GAN blend	0.318
cGAN alpha blend	0.879
cGAN pyramid blend	0.868
cGAN GAN blend	0.846

TABLE II: FID performance- lower scores are better. Our methods outperform the physics-based computer graphics pipeline. Cy-R stands for CyGAN-Render blend, and c-R stands for cGAN-Render blend.

	FID ↓		
Method	Cityscapes[45]	KITTI[50]	ADE20K[51]
Only render [23]	231.768	285.222	361.496
Proposed			
Cy-R alpha blend	175.832	220.223	272.069
Cy-R pyramid blend	196.911	228.277	279.704
Cy-R GAN blend	194.191	234.087	266.615
c-R alpha blend	188.809	220.161	272.877
c-R pyramid blend	202.120	214.488	265.603
c-R GAN blend	194.898	217.663	260.404

rendering method produced unrealistic trees and vegetation (shown with a yellow circle) around the vanishing point of the image. At the same time, the proposed method and the only-cGAN approach generated a more blended background scene with vegetation at the same spot.

On the other hand, only using a GAN-based image synthesizer reduces control over the appearance of objects of interest, such as cars. In Figure 4, the only GAN-based approach failed to generate road-markings. In addition, the surface quality of cars (shown with a red circle) was much lower than the full-render approach and the proposed method. Figure 5 demonstrates the unrealistic shadows of full-render and incomplete vehicle shapes of the only GAN approach with a yellow and red circle. The proposed method alleviates these issues. These results indicate a qualitative validation of our hypothesis: the proposed approach, blending GAN-based synthesizers with conventional rendering, has the best of both worlds.

2) Quantitative results: Figure 6 shows Inception V3 feature vector correlations. Even though real-world images of the Cityscapes and KITTI datasets are entirely different, they have similar latent feature correlations. However, this pattern does not emerge with a synthetic dataset of low visual fidelity. The proposed blended synthetic dataset has, albeit being weak, a similar correlation pattern. In comparison, the same pattern does not emerge with the conventional render or pure GAN approaches. This shows that the proposed blending approach is a good strategy for the realistic representation of driving scenes.

IS, FID, and semantic retention scores are given in Table I, Table II and Figure 7. These scores indicate that the proposed hybrid blending approach consistently outperformes

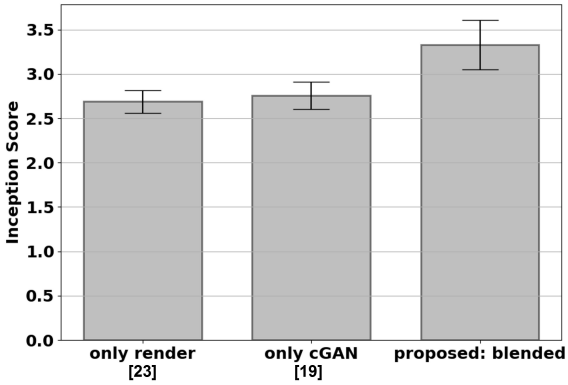


Fig. 7: Inception score [54]. A high Inception score indicates better image quality and higher diversity.

conventional rendering and pure generative adversarial image synthesis.

The Cityscapes dataset contains only urban driving scenes, while ADE20K also has miscellaneous scenes. All of our virtual 3D scenes were in an urban environment. As such, most of the methods received better FID scores for the Cityscapes dataset, as can be seen in Table II.

GAN blend and Alpha blend showed similar performances as shown in Table I and Table II. However, it should be noted that the blending GAN was not trained on an urban driving dataset. The blending performance can possibly be increased with a better blending dataset for training the blending GAN.

The cGAN variants performed better on average as expected, as shown in Table I and II. The synthesized images were both realistic and loyal to the initial semantic layout. However, cGAN requires a paired dataset for training. The full render is better at semantic retention than Cy-GAN variants, but Cy-GAN variants have a higher FID score than rendering. This means that Cy-GAN can generate realistic images but fails to retain the semantic constraints.

VI. Conclusions

This work introduced and investigated the feasibility of Hybrid Generative Neural Graphics (HGNG). The proposed approach utilizes a GAN-based image synthesizer to remove the need for rendering calculations and labor-intensive texture-making steps for background elements while increasing photorealism. In addition, our method achieves full control over the appearance of objects of interest using partial-rendering. Our novel image formation strategy blends the GAN-generated background image with these partial renders and outperforms conventional approaches. Experimental results indicate that conventional driving simulation graphics now have a strong alternative.

In order to train the cGAN-based synthesizers, real-world urban images and their semantic labels, i.e., a paired dataset, are needed. Therefore, with the publication of more paired real-world datasets, the performance of the proposed method can be further increased. On the other hand, CyGANs remove this paired dataset requirement with the use of cycle consistency, but

cyGANs do not perform as well as cGANs. As such, without a paired dataset, the proposed system cannot outperform the conventional pipelines yet. However, potential future developments in domain adaptation and cycle consistency can greatly benefit HGNG and may remove the paired dataset requirement in the future.

This work focused on frame-by-frame image formation with GANs. However, computer graphics applications such as driving simulations may require more temporally consistent approaches. Each subsequent frame of a driving simulation needs to be consistent with the overall sequence. The proposed method already achieves temporal consistency for objects of interest using partial rendering. The temporal consistency of the GAN-generated background scene can potentially be increased with larger urban video datasets. To this end, future work can focus on creating better urban video datasets and developing GAN-based video-to-video synthesis methods.

ACKNOWLEDGMENT

Material reported here was supported by the United States Department of Transportation under Award Number 69A3551747111 for the Mobility21 University Transportation Center. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the United States Department of Transportation.

REFERENCES

[1] V. Punzo and B. Ciuffo, "Integration of driving and traffic simulation: Issues and first solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 354–363, 2010.

[2] A. Lanatà, G. Valenza, A. Greco, C. Gentili, R. Bartolozzi, F. Bucchi, F. Frendo, and E. P. Scilingo, "How the autonomic nervous system and driving style change with incremental stressing conditions during simulated driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1505–1517, 2014.

[3] D. Ludl, T. Gulde, and C. Curio, "Enhancing data-driven algorithms for human pose estimation and action recognition through simulation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21.

[4] S. Minhas, A. Hernández-Sabaté, S. Ehsan, and K. D. McDonald-Maier, "Effects of non-driving related tasks during self-driving mode," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[5] M. Aramrattana, T. Larsson, C. Englund, J. Jansson, and A. Nåbo, "A simulation study on effects of platooning gaps on drivers of conventional vehicles in highway merging situations," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–7, 2020.

[6] W. Yang, L. Zheng, Y. Li, Y. Ren, and Z. Xiong, "Automated highway driving decision considering driver characteristics," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[7] U. Ju, L. L. Chuang, and C. Wallraven, "Acoustic cues increase situational awareness in accident situations: A vr car-driving study," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 2350–2359, 2020.

[8] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 1986, pp. 143–150.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[10] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2018.

[11] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *Stat*, vol. 1050, p. 21, 2018.

[12] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5907–5915.

[13] ——, "Stackgan++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1947–1962, 2018.

[14] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Visualizing and understanding generative adversarial networks," in *International Conference on Learning Representations*, 2019.

[15] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[16] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *International Conference on Learning Representations*, 2018.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[18] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1511–1520.

[19] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.

[20] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8808–8816.

[21] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

[22] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 1144–1156.

[23] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[24] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[25] B. Karis and E. Games, "Real shading in unreal engine 4," *Proc. Physically Based Shading Theory Practice*, vol. 4, 2013.

[26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[27] P. M. van Leeuwen, C. Gómez i Subils, A. Ramon Jimenez, R. Happee, and J. C. de Winter, "Effects of visual fidelity on curve negotiation, gaze behaviour and simulator discomfort," *Ergonomics*, vol. 58, no. 8, pp. 1347–1364, 2015.

[28] X. Zhao and W. A. Sarasua, "How to use driving simulators properly: impacts of human sensory and perceptual capabilities on visual fidelity," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 381–395, 2018.

[29] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[30] R. Fan, X. Wang, Q. Hou, H. Liu, and T.-J. Mu, "Spinnet: Spinning convolutional network for lane boundary detection," *Computational Visual Media*, vol. 5, no. 4, pp. 417–428, 2019.

[31] T. H. Nguyen-Phuoc, C. Li, S. Balaban, and Y. Yang, "Rendernet: A deep convolutional network for differentiable rendering from 3d shapes," in *Advances in Neural Information Processing Systems*, 2018, pp. 7891–7901.

[32] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7708–7717.

[33] I. Gkioulekas, A. Levin, and T. Zickler, "An evaluation of computational imaging techniques for heterogeneous inverse scattering," in *European Conference on Computer Vision*. Springer, 2016, pp. 685–701.

[34] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3907–3916.

[35] M. M. Loper and M. J. Black, "Opendr: An approximate differentiable renderer," in *European Conference on Computer Vision*. Springer, 2014, pp. 154–169.
[36] M. Wang, G.-Y. Yang, R. Li, R.-Z. Liang, S.-H. Zhang, P. M. Hall, and S.-M. Hu, "Example-guided style-consistent image synthesis from semantic labeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1495–1504.
[37] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *33rd International Conference on Machine Learning*, 2016, pp. 1060–1069.
[38] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 994–1003.
[39] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3722–3731.
[40] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 469–477.
[41] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.
[42] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.
[43] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing*. CRC Press, 2010.
[44] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
[45] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
[46] T. Mertens, J. Kautz, and F. Van Reeth, "Exposure fusion: A simple and practical alternative to high dynamic range photography," in *Computer Graphics Forum*, vol. 28, no. 1. Wiley Online Library, 2009, pp. 161–171.
[47] H. Wu, S. Zheng, J. Zhang, and K. Huang, "Gp-gan: Towards realistic high-resolution image blending," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2487–2495.
[48] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays, "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014.
[49] 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 *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.
[50] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
[51] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 633–641.
[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.
[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.
[54] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

Ekim Yurtsever (Member, IEEE) received his B.S. and M.S. degrees from Istanbul Technical University in 2012 and 2014 respectively. He received his Ph.D. in Information Science in 2019 from Nagoya University, Japan. Since 2019, he has been with the Department of Electrical and Computer Engineering, The Ohio State University as a research associate.

His research focuses on artificial intelligence, machine learning, computer vision, reinforcement learning, intelligent transportation systems, and automated driving systems.

Dongfang Yang received his bachelor's degree in microelectronics from Sun Yat-sen University, Guangzhou, China, in 2014. He has been with The Ohio State University since 2015 and received his Ph.D. in Electrical and Computer Engineering from The Ohio State University in 2020. He is currently a graduate research associate at The Ohio State University. His research interests include control systems, computer vision, and machine learning with applications in intelligent transportation and autonomous driving.

Mert Koc received his bachelor's degree in Electrical and Electronics Engineering from Middle East Technical University (METU), Ankara, Turkey, in 2018. He has been with The Ohio State University since 2018 and is going to receive his MSc. in Electrical and Computer Engineering from The Ohio State University in 2021. He is currently a graduate research associate at The Ohio State University. His research interests include computer vision, robotics and machine learning with applications in autonomous driving.

Keith A. Redmill (S'89–M'98–SM'11) received the B.S.E.E. and B.A. degrees in mathematics from Duke University, Durham, NC, USA, in 1989 and the M.S. and Ph.D. degrees from The Ohio State University, Columbus, OH, USA, in 1991 and 1998, respectively. Since 1998, he has been with the Department of Electrical and Computer Engineering, The Ohio State University, initially as a Research Scientist. He is currently a Research Associate Professor. He is a coauthor of the book *Autonomous Ground Vehicles*. He has significant experience and expertise in intelligent transportation systems, intelligent vehicle control and safety systems, sensors and sensor fusion, wireless vehicle to vehicle communication, multi-agent systems including autonomous ground and aerial vehicles and robots, systems and control theory, virtual environment and dynamical systems modeling and simulator development, traffic monitoring and data collection, and real-time embedded and electromechanical systems.