36th European Modeling & Simulation Symposium, 012
21st International Multidisciplinary Modeling & Simulation Multiconference

2724-0029 © 2024 The Authors.
doi: 10.46354/i3m.2024.emss.012

Enhancing Photorealism of Physically Accurate 3D Simulated Images Using GANs

Veronica Campana[1,*], Jorge Luis Jiménez Aparicio[2], Jürgen Roßmann[2] and Ruggero Carli[1]

[1] Department of Information Engineering, University of Padova, Via Gradenigo 6/B, Padova 35131, Italy
[2] Institute for Man-Machine-Interaction, RWTH Aachen University, Ahornstrasse 55, Aachen 52074, Germany

[*] Corresponding author. Email address: veronica.campana@studenti.unipd.it

Abstract

Reducing the realism gap between real and simulated sensor data remains a critical challenge in current research in modeling and simulation. Virtual Testbeds (VTBs) provide a safe, cost-effective environment for research, development, and testing but often fall short in the photorealism of the simulated camera sensors compared to real ones. To address this, we leveraged a state-of-the-art framework based on Generative Adversarial Networks (GANs) to enhance the photorealism of these VTBs. This paper introduces a generative Artificial Intelligence (AI) framework originally proposed to translate simulated image data from video games into photorealistic urban scenes. We adapt and extend this framework to different simulated data generated with a physically accurate 3D simulator. Various implementations with urban scenes were proposed and analyzed to assess their effectiveness in real-world scenarios. We evaluated our promising implementations using a real-time capable object detector to assess the impact of the enhancements and to identify persistent problems in enhancing the realism of simulated data.

Keywords: Image-to-image translation; photorealism enhancement; Generative Adversarial Networks; simulation environments; Virtual Testbeds

1. Introduction

In recent years, progress in the field of deep learning (DL), has led to the use of DL frameworks in applications involving style adaptation between different visual domains. Mapping an image from one domain to another using DL techniques is known as image-to-image translation, while its video counterpart is addressed as video-to-video synthesis. One common approach in this field involves using AI frameworks based on Generative Adversarial Networks (GANs). These models are implemented such that, the initial data are modified to preserve their semantic content while being as visually similar as possible to the target reference domain.
One of the potential applications of the image-to-image translation and video-to-video synthesis frameworks is to improve the photorealism of simulated images, which can be used to significantly enhance the visual realism of simulation environments and generate closer-to-reality virtual training data. The use of realistic Virtual Testbeds (VTBs) has become particularly important in the last years. This is because it allows us to work in environments where the gap between simulation and real-world is minimized (Müller et al., 2021). These environments can be more practically useful and efficient for AI applications deployed in real-world scenarios. Moreover, employing DL frame-

works can reduce the computational and human effort typically required to create accurate and realistic simulation environments, and overcome the challenge of directly modeling the geometric and material features of the system, exploiting Neural Networks trained on large datasets (Richter et al., 2021).
The work proposed in this paper starts from the analysis and implementation of the promising framework "Enhancing Photorealism Enhancement" (EPE) presented in (Richter et al., 2021), introduced for image-to-image translation and video-to-video synthesis of simulated images extracted from the GTA V videogame, to achieve urban scenes with an enhanced photorealism. This paper aims to adapt the framework to work with simulated sensor data generated within a VTB in the accurate 3D simulation software VEROSIM (VEROSIM GmbH) to enhance photorealism in the simulation environment. This could facilitate the generation of photorealistic virtual training data that further closes the gap to real data. Future research will validate the benefit of training with such data, but a performance improvement with respect to AI trained with lower quality virtual data is expected, particularly in AI intended for deployment in real-world sectors where annotated open real datasets are scarce or expensive and even dangerous to create (e.g. construction, forestry, agriculture, space).

2. State of the art and Related works

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) were first introduced in (Goodfellow et al., 2014), to propose an innovative generative model. Unlike discriminative models, generative models are trained to learn the statistical distribution of a given training dataset to generate new samples from the learned distribution (Creswell et al., 2018).
GANs are based on the presence of two neural networks, a generator, and a discriminator, that work in an adversarial manner. In the learning strategy introduced by GANs, the generator aims to generate synthetic samples that are as similar as possible to the ones of a target training distribution, without having access to it. On the other side, the goal of the discriminator is to learn how to correctly discriminate samples if they are coming from the real target distribution or if they are samples generated by the generator network. The adversarial learning procedure leads both networks to improve their outcomes and the final ideal result consists of having the generator generating examples that are indistinguishable from the ones sampled from the training dataset (Goodfellow et al., 2014).
Important extensions to GANs are Conditional GANs (CGANs), introduced in (Mirza and Osindero, 2014) and infoGANs presented in (Chen et al., 2016).
Due to their significant achievements in generating realistic data samples, GANs are increasingly employed in a wide range of applications, such as image generation, data augmentation, style transfer and in numerous other domains (Chakraborty et al., 2023).

2.2. Image-to-image translation and video-to-video synthesis deploying GANs

The primary aim of image-to-image translation tasks is to generate images that assume the style of the data belonging to the target domain and appear indistinguishable from them. In parallel to it, its counterpart that addresses video data is known as video-to-video synthesis. In this case, the additional objective is to preserve also the temporal continuity among the consecutive frames of the video (Wei et al., 2018).
Many tasks related to image processing and computer graphics can be addressed as image-to-image translation and video-to-video synthesis tasks. Some examples of these tasks are colorization of images (Zhang et al., 2016), translation from low resolution to high resolution (Lee et al., 2018), obtaining photorealistic images starting from semantic label maps (Isola et al., 2018) or from edge maps. In addition to these, the task of enhancing the photorealism of simulated images assumes a significant importance.
In (Zhu et al., 2017) and in (Hoffman et al., 2017), the problem of unsupervised image-to-image translation is addressed by employing adversarial networks and an additional cycle-consistency loss to preserve consistency among the starting and generated samples and to ensure to construct a reversible map. Differently from (Zhu et al., 2017), in (Hoffman et al., 2017) a further semantic loss is added to improve the translation. Also in (Huang et al., 2018), cycle-consistency is imposed but, in this case, it is assumed that a content latent code, that is domain-invariant and a style latent code, characteristic of the specific domain, can be associated with each image and it considers the additional aim of capturing the diversity of the different target domains. Finally, in (Park et al., 2020) the idea of employing contrastive learning is introduced to face the task of unsupervised image-to-image translation.
Regarding the video counterpart, different methods are proposed to address the task of video-to-video synthesis, with the further aim of ensuring temporal consistency among consecutive frames (Saha and Zhang, 2024). For instance, (Bashkirova et al., 2018) treat the inputs and outputs as three-dimensional tensors and propose a 3D GAN-based method to address the domain translation by exploiting also the information encoded in the temporal dimension. In (Chen et al., 2019) and in (Park et al., 2019) previously proposed image-to-image translation approaches are extended to deploy additional methods based on using the optical flow to ensure consistent motion translation among consecutive frames. In (Rivoir et al., 2021), the aim of obtaining photorealistic videos is addressed by leveraging image translation strategies combined with methods that guarantee temporal consistency by exploiting texture and appearance information and enforcing view consistency.
In EPE, the authors introduced a promising work to en-

hance the photorealism of simulated images and videos. The EPE framework shows relevant results in obtaining significant photorealism enhancement in unsupervised domain translation and in eliminating possible scene artifacts. The practical implementations described in this current work are based on the adaptation of the model proposed in EPE. A significant strategy used in their work consists of exploiting additional information related to the geometry, materials, and lighting of the objects in the scene, which is extracted from the rendering pipeline of the video game. This additional information is associated with each input simulated image and is concatenated in a so-called G-buffers file. In the framework, the G-buffers are used in the network, managing them with a G-buffers encoder that, using masks derived from the ground truth semantic label maps, processes this additional information differently for different object classes. For example, sky regions contain only the extra information about lights, and not about geometry or material. Additionally, to reduce scene artifacts, typically encountered when implementing previous image-to-image translation strategies, the authors propose to analyze the distribution of objects in images from different datasets and to use specific sampling techniques that take into account these differences. To translate images from the simulated domain to the real-world domain, a perceptual discriminator is used during the training procedure to distinguish the enhanced images from the real-world ones. An additional metric is employed to penalize substantial structural differences between the simulated and enhanced images. To enable the discriminator to evaluate the photorealism of the simulated images at a high level, the authors propose considering not only a binary real vs. fake decision but also adding other classification objectives. The discriminator is trained to evaluate multiple perceptual feature maps and is integrated with semantic information, obtained by leveraging the pre-trained semantic segmentation network MSeg (Lambert et al., 2021).

3. Materials and Methods

The practical implementations presented in this work involved using the EPE framework to enhance the photorealism of different simulated input data.
Specifically, four different practical implementations were considered. For each one of them, the framework was first trained from scratch to learn the GANs model, and subsequently, inferences were obtained using the weights achieved during the training phase.
The framework was trained using two input datasets: a simulated image dataset and a real-world image dataset. To create the simulated dataset, each simulated RGB image was associated with a ground truth semantic label map, a robust label map generated with the MSeg pre-trained model, and a set of G-buffers data. On the other hand, the real-world image dataset was extracted from the Cityscapes dataset (Cordts et al., 2016). Each real-world image in this dataset was associated with the corresponding robust label map generated using the MSeg pre-trained model.
For each implementation, both the simulated and the target datasets used to train the framework consist of approximately 20K samples with a resolution of 960x540.
After training the model, inferences were obtained from a separate simulated test dataset consisting of 975 consecutive frames of a drive in the VTB. This dataset had the same structure as the training simulated dataset.
Among the four different simulated datasets used to train the model, the first one was created with simulated RGB images extracted from the dataset "Playing for data: Ground truth from computer games" (Richter et al., 2016), which was created using the GTA V video game. The latter three contained data generated with the physically accurate 3D simulator VEROSIM (VEROSIM GmbH). The details of the simulated datasets used in the implementations are summarized in the following subsections. For each dataset, the code of the EPE framework was modified to adapt to different G-buffers information when considering them as input to the model.

3.1. Implementation with simulated images from GTA V videogame

The purpose of this initial implementation was to utilize simulated images similar to those used in the EPE reference paper, to achieve comparable photorealism enhancement results. Similar to EPE, the simulated images and ground truth label maps were extracted from the dataset introduced in (Richter et al., 2016), which is generated from the GTA V videogame. However, differently from EPE, the G-buffers data were created with the pre-trained AI model Omnidata (Eftekhar et al., 2021), since access to the rendering engine of the GTA V videogame was restricted, impeding the extraction of ground truth data. Due to this limitation, in this implementation only G-buffers encoding the information on the structure and geometry of the objects in the scene were employed. Namely, the depth map and the surface normals map were associated with the input simulated images. Thus, the primary difference between this practical implementation and the one described in EPE was related to the G-buffers data used. In Figure 1, an example of the starting simulated RGB image, with the corresponding G-buffers data is shown.
In this implementation, the information in the G-buffers file was equally assigned to all objects in the scene, without distinguishing among the different semantic classes. In this case, the code of the framework EPE was modified to take as input G-buffers with 6 channels associated with each pixel of the corresponding simulated image.

3.2. Implementation with simulated images generated with VEROSIM

In the three different implementations presented in this subsection, the framework was trained using different

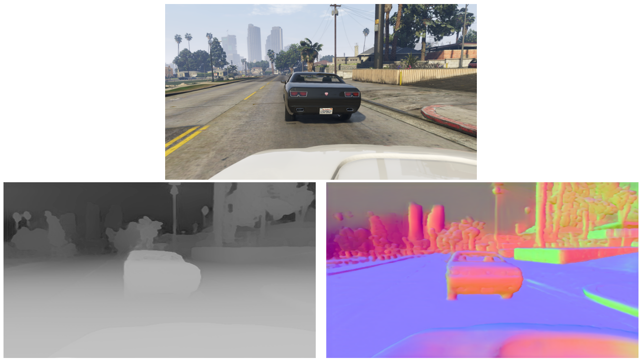

Figure 1. Simulated RGB image extracted from the dataset (Richter et al., 2016) (top), depth map generated with the model (Eftekhar et al., 2021) (left), surface normals map generated with the model (Eftekhar et al., 2021) (right).

Figure 2. Ray-tracing simulated color image (left), corresponding ground truth depth map (middle) and ground truth surface normals map (right), all generated with the VTB in VEROSIM.

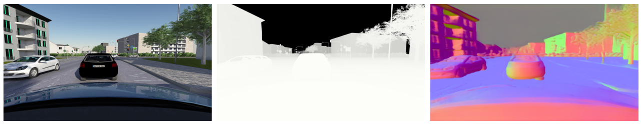

Figure 3. Ray-tracing simulated color image (left), and corresponding ground truth depth map (middle), both generated with the VTB in VEROSIM, and surface normals map generated with the AI model (Eftekhar et al., 2021).

simulated data extracted from simulation videos of urban street scenarios generated within a VTB in VEROSIM. The simulation data contained objects such as cars, buildings, road signs, traffic lights, and vegetation that simulate realistic street views. However, it was observed that the dataset generated with VEROSIM had a lower variety of street scenes compared to the dataset extracted from the images generated with the video game GTA V.
In all three datasets used to train the framework, the simulated RGB images employed were ray-tracing color images. The differences between the various simulated datasets were determined by the different G-buffers information utilized. The data collected in the G-buffers files, which were used as input to the framework, contained the following characteristics for each one of the three implementations that were considered:

1. The G-buffers file contained the ground truth depth map and surface normals map generated with VEROSIM, as shown in Figure 2. A key difference in this implementation compared to the previous one is how the surface normals were oriented. Previously, the surface normals were aligned with the frame of the sensor attached to the ego-vehicle; however, in this implementation, they were defined and oriented with respect to the world frame of the simulation environment. As a result, when the ego-vehicle changes its orientation with respect to the world frame, the color codes associated with the x, y, and z normal vectors change accordingly. Therefore the color code used for the surface normal maps was not consistent among the different frames of the training dataset.
2. The G-buffers file contained the ground truth depth map generated with VEROSIM and the surface normals map generated using the pre-trained AI model Omnidata, as can be seen in Figure 3. Therefore the only difference with respect to the dataset generated for the previous implementation is related to the surface normals information. In this case, the surface normals maps were defined with respect to the reference frame of the sensor attached to the ego-vehicle, as in the implementation with GTA data.
3. The G-buffers file contained the ground truth depth map, the ground truth surface normals map, and the ground truth albedo map, all generated by employing VEROSIM, as shown in Figure 4. In this simulated dataset, the surface normals maps were defined with respect to the reference frame of the sensor attached to the ego-vehicle. This allowed a consistent color code used for the surface normals maps throughout all the frames in the dataset.

In the first two implementations using simulated data generated with VEROSIM, the G-buffers information was equally assigned to all objects of the scene, without differentiating between the various semantic classes. Instead, in the third implementation with VEROSIM simulated images, the albedo maps were also used and they were the only data associated with the sky semantic class. Therefore, for this class only the information related to the color, contained in the albedo G-buffers, was considered, while the information related to the structure and the geometry, not relevant for the sky region, is completely excluded from this semantic class. The aim of this implementation choice was to address the presence of some artifacts encountered in the sky regions in the previous implementations.

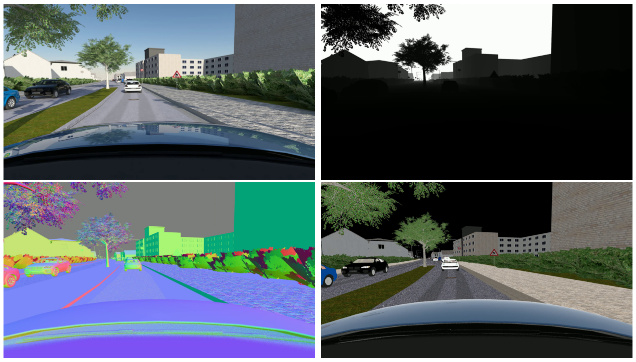

Figure 4. Ray-tracing simulated color image (top-left), corresponding ground truth depth map (top-right), ground truth surface normals map (bottom-left), and ground truth albedo map (bottom-right), all generated with the VTB in VEROSIM.

For the first two implementations described in this section, the code of the EPE framework was modified to take 6-channel G-buffers as input for each pixel of the simulated image. In the third implementation, 9-channel G-buffers were used.

4. Results and Discussion

This section presents the details of the training procedure and the resulting inferences obtained implementing the framework with the datasets described in subsections 3.1 and 3.2.
As indicated in the EPE paper, both the generator and the discriminator were trained with an L2 loss, the Adam optimizer was employed in training both networks with weights decay 0.0001 and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate used was 0.0001, halved every 100K iterations and the batch size was set to 1. The NVIDIA GeForce RTX 3090 GPU was used for training the model and obtaining inference in all the implementations considered.
In each implementation, the framework was trained for 1M iterations. Inferences were obtained considering the framework fully trained. In some cases, inferences were also obtained by considering the model learned after a lower number of iterations. In each implementation, inferences were obtained by using a test dataset that corresponds, in the way it is generated, to the datasets used during the training phase of the framework. Additionally, some cross-inferences were obtained. Moreover, the corresponding video data were generated from the inferences obtained in each implementation.
All the inferences obtained were intended to replicate the real-world images from the Cityscapes dataset (Cordts et al., 2016). Initially, the inferences are evaluated qualitatively, checking the overall photorealism of the scene, the presence of any artifacts and the preservation of object structures for each set of resulting inferences. Furthermore, the temporal continuity of generated videos from the inference frames is considered.

Finally, results that show potential for improvement in either simulation time or photorealism are processed using the real-time capable object detector YOLOv3 (Redmon and Farhadi, 2018) employing the standard weights provided by its authors. The evaluation is done using the standard metrics: Average Precision (AP) and mean Average Precision (mAP) (Everingham et al., 2010).

4.1. Inferences from the model trained on GTA V simulated images

The inferences presented in this section were obtained employing the simulated dataset described in subsection 3.1.
In Figure 5, an example of the inferences obtained by the model trained for 1M iterations is shown and compared to the corresponding starting simulated image. The resulting inferences are significantly more photorealistic when compared to the images extracted from the starting dataset of GTA V frames (Richter et al., 2016). Specifically, the resulting enhanced images reveal smooth and more realistic asphalt, similar to one of the streets shown in the Cityscapes dataset. The vegetation is also far more voluminous and photorealistic and gloss has been added to the vehicles to resemble the vehicles of real-world images. No significant artifacts are encountered in the obtained inferences.

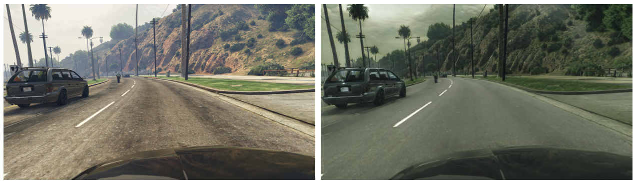

Figure 5. Starting simulated image from GTA V (left) compared to the corresponding resulting inference obtained with the model trained for 1M iterations (right).

4.2. Inferences from the model trained on simulated images generated with VEROSIM

This subsection presents the inferences obtained from the model trained with the datasets described in subsection 3.2. Therefore, the objective of using the framework is to evaluate the photorealism enhancement in simulated images generated with the VTB in VEROSIM. The inferences obtained for each one of the three simulated datasets introduced in subsection 3.2 were evaluated in the same order:

1. Firstly, the inferences obtained with the framework implemented with the first VEROSIM simulated dataset are considered. As can be observed in Figure 6, in the resulting inference obtained with the weights of the model trained for 1M iterations, all the structures of the objects are preserved and the photorealism of the single objects is enhanced in terms of visual style. This can be seen for example from the asphalt of the street that is smoother, from the vegetation that is more voluminous, and from the majority of objects in the scene, that are rendered in such a way to resemble the ones of the Cityscapes dataset. However, the artifacts encountered, especially on the ego-vehicle and on the sidewalk, decrease significantly the overall photorealism of the scene. In Figure 6, the corresponding inference obtained with the model trained for 900K iterations, it can be seen that, even if some artifacts are encountered, they are significantly less evident than the ones observed in the inference obtained with the model trained for 1M iterations. Comparing the visual style, the inference obtained with the model trained for 900K iterations is characterized by colors slightly less similar to the ones of the Cityscapes dataset than the ones obtained with the model trained for 1M iterations, but, thanks to the sig-

nificant fewer artifacts, the inferences obtained with the model trained for 900K iterations, have more overall realistic features. Considering the resulting videos obtained with the corresponding inferences, they show a discrete temporal consistency in most of the scene but some temporal flickering artifacts are encountered mainly on the sidewalks and ego-vehicle.

2. In this case, the inferences were obtained by implementing the framework with the second VEROSIM dataset presented in the previous section. Both the inferences obtained with the model trained for 900K iterations and 1M iterations are shown in Figure 7. As in the previous case better overall photorealism can be observed by evaluating the inferences obtained with the model trained for 900K iterations. Specifically, in the inference obtained with the model trained for 900K iterations, it can be seen that the general photorealism of the scene is enhanced with respect to the starting simulated image and all the elements of the scene show a visual style more similar to the one of the reference Cityscapes images (Cordts et al., 2016). In the inferences obtained with the model trained for 900K iterations, only small artifacts are encountered especially in the ego-vehicle, while more evident artifacts are visible in the sample inference obtained with the model trained for 1M iterations. Also in this case, the videos generated from the corresponding inferences show quite good temporal stability, and only small artifacts can be observed in the sidewalks.

3. Finally, the inferences obtained implementing the framework with the third VEROSIM dataset are presented. In the inference obtained with the model trained for 1M iterations shown in Figure 8, an enhanced photorealism of all the objects of the scene can be observed, additionally, the visual style is noticeably more similar to that of the real-world reference images. Though there are some small artifacts present for example in the sidewalk and in the lower part of the ego-vehicle, they are much less visible than the ones in the inferences of the previous implementations with simulated data generated with VEROSIM. Moreover, the video generated with the corresponding inferences shows good temporal continuity and only insignificant temporal artifacts are observed.

In all the inferences obtained, all the structures of the objects of the scene are well preserved.

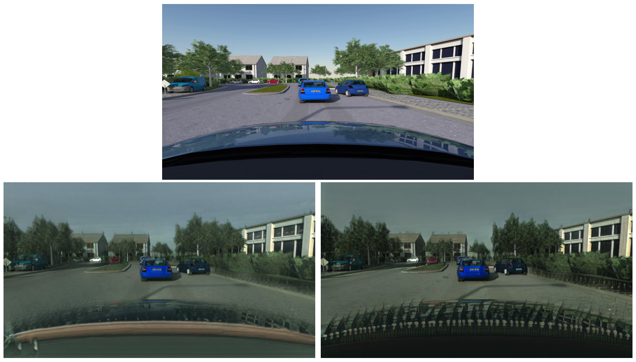

Figure 6. First implementation with VEROSIM data: starting ray-tracing simulated image (top) compared to the corresponding resulting inferences obtained with the model trained for 900K iterations (bottom-left) and 1M iterations(bottom-right).

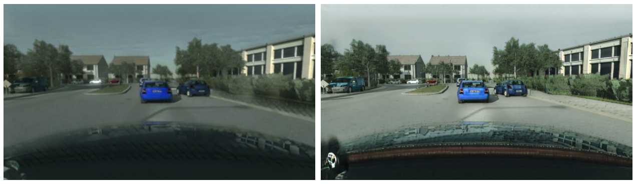

Figure 7. Second implementation with VEROSIM data: resulting inferences obtained with the model trained for 900K iterations (left) and 1M iterations (right).

In addition, some sets of cross inferences were obtained. These inferences were obtained from a simulated test dataset that was from a different domain as that of the of the simulated dataset used to train the model. The first set of cross-inferences used the weights from the model trained for 1M iterations which yielded the best results (third implementation with the VEROSIM dataset). The inferences were obtained starting from a simulated test dataset of raster images, which were generated with VEROSIM. Raster images are generated with a raster camera sensor and are lower-quality images with respect to the ray-tracing images previously considered. The resulting inference obtained from the simulated raster image can be seen in Figure 9.

For the second set of cross-inferences, the weights from the model trained for 1M iterations using the GTA V simulated dataset were employed, while the same test dataset used to test the best VEROSIM implementation, presented in subsection 3.2, was used. Due to the difference between the weights of the trained models, the albedo maps were not used in this case. Figure 10 shows the resulting inference obtained from the second set of cross-inferences.

As can be observed, in both cases the resulting cross-inference shows more photorealistic features with respect to the starting simulated images, in particular in terms of rendered vegetation, buildings and vehicles. However, the photorealism enhancement in the output images is less evident compared to that of the inferences processed without any cross-inference, as expected. This can be justified by the fact that when running cross-inferences, it is necessary to generalize to objects in the scene that differ a lot, in terms of colors and visual style, from the ones of the training dataset. This exposes the difficulties that this framework has to change domains without additional fine-tuning or Transfer Learning approaches.

4.3. Object detection

Using the standard metric AP with intersection over union (IoU) thresholds of 50% (AP@0.5) and 75% (AP@0.75), we calculated the mAP to evaluate the expected performance of the object detector YOLOv3 across both the base and enhanced test datasets. Despite the detector's relatively

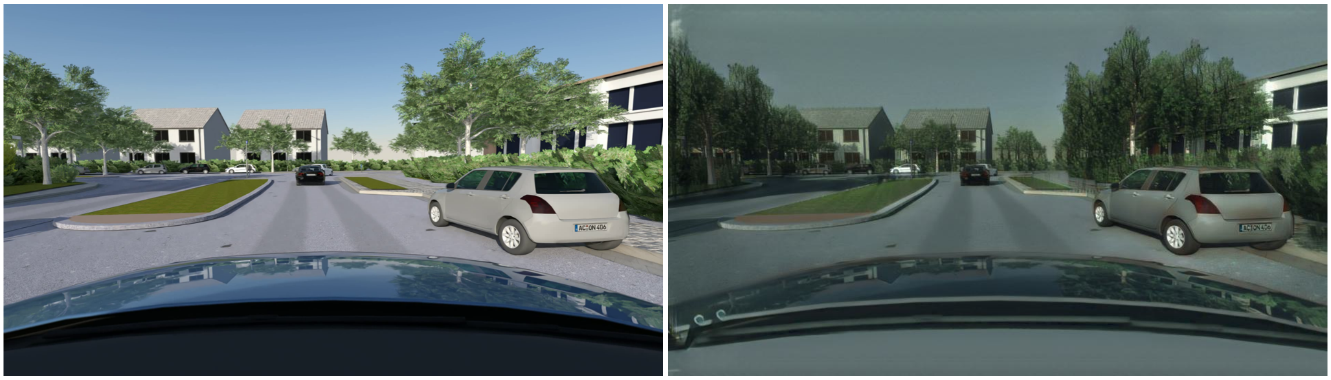

Figure 8. Third implementation with VEROSIM data: starting ray-tracing simulated image (left) compared to the corresponding resulting inference obtained with the model trained for 1M iterations(right).

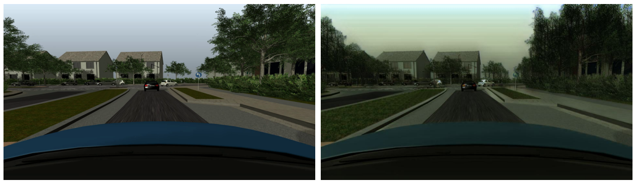

Figure 9. Cross-inferences obtained from the raster dataset: starting simulated raster image (left) compared to the corresponding resulting inference obtained with the model trained for 1M iterations (right).

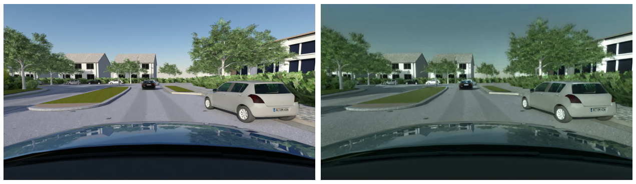

Figure 10. Cross-inferences obtained with the model trained with GTA V dataset: starting simulated image (left) compared to the corresponding resulting inference obtained with the model trained for 1M iterations (right).

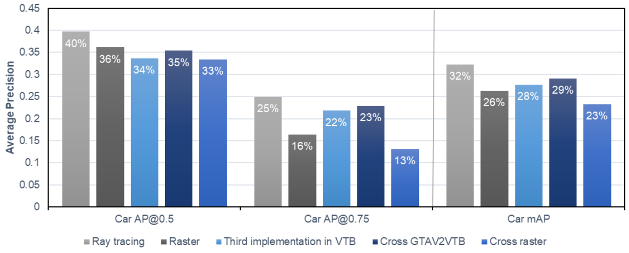

Figure 11. Average Precision (AP) and mean AP (mAP) for car detections using the object detector YOLOv3 for five selected datasets: two base simulated ones and three enhanced ones, two of which are the cross-inference datasets. Intersection over Union (IoU) of 50% and 75%.

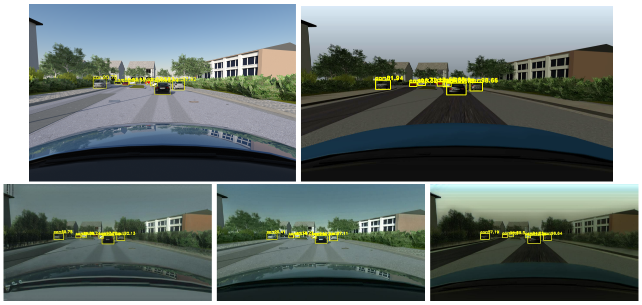

Figure 12. Car detection examples for the five evaluated datasets. From top to bottom and left to right in each row: ray tracing, raster, third implementation with the VTB in VEROSIM, cross-inference obtained with the model trained with GTA V, and cross-inference starting from simulated raster images datasets.

low performance of the detector, which can be attributed to the challenging nature of the dataset (pixel-accurate ground truth labels given by VEROSIM, regardless of object size) and the discrepancy between the training and evaluation domains of YOLO, this metric is valuable for comparing our results and identifying potential issues for future research.

As shown in Figure 11, the base ray tracing dataset exhibits the best performance. The enhanced datasets show lower performance, with the cross-inference using the model trained with GTA V data to enhance the VTB being the best among them. As expected, performance generally decreases at the AP@0.75 threshold, especially in the raster and cross inference of the raster datasets. This indicates that improving the quality of the dataset (ray tracing vs raster) increases performance in more challenging detection tasks. However, the evaluation also shows that enhanced visual realism does not necessarily translate to improved object detection performance. This suggests that the artifacts introduced by the enhancements significantly impact real AI tasks, and enhancement efforts should focus not only on visual improvement but also on preserving the structural integrity of the objects in the scene.

Figure 12 displays the first frame from the five selected datasets showing the car detections. It can be observed that both the enhanced and low-quality datasets struggle more with detecting smaller objects.

5. Conclusions

The primary goal of this research was to adapt and apply the framework introduced in "Enhancing Photorealism Enhancement" (EPE) to generate photorealistic simulated sensor data from simulation data obtained from a Virtual Testbed (VTB) available in the physically accurate 3D simulator VEROSIM. Our work demonstrated several implementations of the framework, each tailored to different features of simulated data. Initially, we achieved results comparable to those presented in the original EPE research, using a reduced amount of G-buffers information. Subsequently, in the first two implementations with VEROSIM data, the inferences obtained show enhanced photorealism in their visual style, with respect to the corresponding starting simulated images, but some artifacts visible in the scene affect the realism of the represented subjects. Notably, in the last implementation with VEROSIM data, the inferences obtained showed enhanced photorealism while further decreasing the amount of artifacts encountered.

Additionally, our cross-inference tests revealed limitations in domain generalization, underscoring the need for potentially incorporating transfer learning or further fine-tuning to enhance model robustness across varied inputs. Object detection tests further illustrated that while enhanced datasets could mimic real-world textures and lighting better, they did not necessarily translate to improved detection performance, particularly under stricter intersection over union (IoU) thresholds.

6. Further works

Our results suggest future research should focus on enhancing visual quality while simultaneously minimizing the introduction of artifacts and improving the generalization capabilities of photorealism enhancement frameworks. It will be crucial to enhance the structural integrity and consistency across various simulation environments, especially for real-world AI applications where accuracy and reliability gain significant importance.

Further research could also explore the inclusion of raster simulated sensor images during the training phase to potentially decrease the computational demands of the VTB. Given that better results were achieved with increased G-buffers information, it stands to reason that expanding the G-buffers data associated with simulated images could yield further improvements. Specifically, integrating additional scene lightning information might enhance the overall results.

References

Bashkirova, D., Usman, B., and Saenko, K. (2018). Unsupervised video-to-video translation.

Chakraborty, T., S, U. R. K., Naik, S. M., Panja, M., and Manvitha, B. (2023). Ten years of generative adversarial nets (gans): A survey of the state-of-the-art.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets.

Chen, Y., Pan, Y., Yao, T., Tian, X., and Mei, T. (2019). Mocycle-gan: Unpaired video-to-video translation.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65.

Eftekhar, A., Sax, A., Malik, J., and Zamir, A. (2021). Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. (2017). Cycada: Cycle-consistent adversarial domain adaptation.

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018). Multimodal unsupervised image-to-image translation.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2018). Image-to-image translation with conditional adversarial networks.

Lambert, J., Liu, Z., Sener, O., Hays, J., and Koltun, V. (2021). Mseg: A composite dataset for multi-domain semantic segmentation.

Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M. K., and Yang, M.-H. (2018). Diverse image-to-image translation via disentangled representations.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Müller, M. G., Durner, M., Gawel, A., Stürzl, W., Triebel, R., and Siegwart, R. (2021). A photorealistic terrain simulation pipeline for unstructured outdoor environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9765–9772.

Park, K., Woo, S., Kim, D., Cho, D., and Kweon, I. S. (2019). Preserving semantic and temporal consistency for unpaired video-to-video translation. *CoRR*, abs/1908.07683.

Park, T., Efros, A. A., Zhang, R., and Zhu, J.-Y. (2020). Contrastive learning for unpaired image-to-image translation.

Redmon, J. and Farhadi, A. (2018). YOLOv3: An Incremental Improvement.

Richter, S. R., AlHaija, H. A., and Koltun, V. (2021). Enhancing photorealism enhancement.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing.

Rivoir, D., Pfeiffer, M., Docea, R., Kolbinger, F., Riediger, C., Weitz, J., and Speidel, S. (2021). Long-term temporally consistent unpaired video translation from simulated surgical 3d data.

Saha, P. and Zhang, C. (2024). Translation-based video-to-video synthesis.

VEROSIM GmbH. Verosim solutions.

Wei, X., Zhu, J., Feng, S., and Su, H. (2018). Video-to-video translation with global temporal consistency. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 18–25, New York, NY, USA. Association for Computing Machinery.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. *CoRR*, abs/1603.08511.

Zhu, J., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593.