QuickRender: A Photorealistic Procedurally Generated Dataset with Applications to Super Resolution (Student Abstract)

Morgan Payette and Charlotte Curtis

Mount Royal University
4825 Mt Royal Gate SW
Calgary, Alberta T3E 6K6 Canada
mpaye629@mtroyal.ca, ccurtis@mtroyal.ca

Abstract

Rendering of complex scenes from software such as Blender is time consuming, but corresponding auxiliary data such as depth or object segmentation maps are relatively fast to generate. The auxiliary data also provides a wealth of information for tasks such as optical flow prediction.

In this paper we present the QuickRender dataset, a collection of procedurally generated scenes rendered into over 5,000 sequential image triplets along with accompanying auxiliary data. The goal of this dataset is to provide a diversity of scenes and motion while maintaining realistic behaviours. A sample application using this dataset to perform single image super resolution is also presented.

The dataset and related source code can be found at https://github.com/MP-mtroyal/MetaSRGAN.

Introduction

Synthetic datasets have proven to be valuable tools for supervised learning tasks, particularly in applications such as optical flow where ground truth is challenging to define. Blender is a popular open source 3D creation suite that can be used to generate images along with auxiliary data. While this data is relatively efficient to generate, rendering realistic images and videos in high resolution is an expensive process, taking upwards of 40 minutes to render a single 4k resolution image on a consumer-grade GPU.

Relatively unrealistic rendered scenes such as FlyingChairs (Dosovitskiy et al. 2015) and FlyingThings3D (Mayer et al. 2016) have proven to be nonetheless effective for training optical flow models. Recent work has demonstrated that synthetic data can be enhanced with learned representations, in particular showing an improvement when atmospheric effects such as fog are introduced (Sun et al. 2021).

This paper presents the QuickRender dataset, rendered as image triplets in Blender together with efficient auxiliary data. The dataset includes a diversity of realistic scenes where all foreground and background objects are fully rendered. We then use the dataset to show preliminary results of using auxiliary data and low-resolution rendered frames to infer higher resolution versions in less time than rendering the full image.

In addition to this example usage, we anticipate that this dataset will be valuable for other tasks such as inter-frame interpolation, optical flow and depth prediction, and more.

Related Work

3D rendered datasets such as FlyingChairs (Dosovitskiy et al. 2015), FlyingThings3D (Mayer et al. 2016), and Virtual KITTI (Gaidon et al. 2016) have been demonstrated to provide effective ground truth for both depth prediction (Maximov, Galim, and Leal-Taixe 2020) and optical flow (Sun et al. 2018). However, these datasets share a common problem of lacking diversity in their scenes, potentially impacting the ability for networks to generalize effectively outside of the test domain. Additionally, datasets like FlyingChairs (Dosovitskiy et al. 2015) neither look photorealistic nor present a scene representative of real-world scenarios. Finally, the majority of existing datasets use static photographic backgrounds for image diversity rather than fully rendered background scenes, resulting in an oversimplified discrepancy between foreground and background objects.

In addition to synthetic datasets, the KITTI (Geiger, Lenz, and Urtasun 2012) self-driving car dataset is commonly used as a benchmark for temporal tasks such as inter-frame interpolation and optical flow prediction. While inherently photorealistic, this dataset contains many similar scenes and image triplets with only small changes. The Vimeo90K (Xue et al. 2019) dataset addresses this by sampling thousands of different videos posted to Vimeo, but it is lacking in auxiliary data.

These types of networks can be used to augment the functionality of other networks, such as DAIN (Bao et al. 2019). When metadata is available to be included into a network's input, a dataset with generated data is required, such as [some paper]. Datasets such as MD dataset (Li and Snavely 2018) or FlyingChairs (Dosovitskiy et al. 2015) present images paired with corresponding metadata via 3D rendering.

The QuickRender Dataset

We present the QuickRender dataset, which provides comprehensive auxiliary data paired to image triplets throughout a diverse range of photorealistic scenes. The scenes used to create QuickRender allow for procedural manipulation of parameters, helping to ensure high levels of di-

Figure 1: Sample images from each scene in the QuickRender dataset

versity between each set of image triplets while maintaining realistic motion of objects, lighting effects, and surface textures. Rather than taking multiple samples of the same video, QuickRender uses a new random seed to generate each triplet to increase diversity.

QuickRender comprises 5 scenes, each of which randomizes the following properties: camera position, lens focal length, position of objects, velocity of objects, where objects are present in the scene, parameters of diffuse shaders, HDRI texture and its parameters, time of day, weather, as well as some parameters that are specific to each scene. A start frame between 1 and 200 within the animation is randomly selected as the first of 3 sequential frames. Sample images from each of the five scenes are shown in figure 1; two are representative of product renderings, while the other three scenes are photorealistic.

In addition to the images the following auxiliary data are normalized and saved as PNGs: object segmentation maps, surface normal vectors, depth maps and velocity maps.

Application to Super Resolution

To demonstrate the capabilities of this dataset and provide an accelerated means of rendering high resolution images from Blender, an augmentation of SRGAN (Ledig et al. 2017) was created, named MetaSRGAN. This version aims to perform single image super resolution using the low resolution rendered image α, as well as the low resolution auxiliary data created during render passes. For this application only depth, object segmentation maps, and surface normals were used. These features were chosen as the most relevant to the super resolution task; inclusion of velocity maps increased the network size without a visible impact on the final result.

The architecture of this network includes a pre-trained SRGAN, which is used to upscale the auxiliary data channels by a factor of 4. The 5 channels (including 3 for surface normals) are concatenated and used as an input to a fully convolutional autoencoder, which is trained from scratch. After training, only the encoder from this network is used, producing a latent representation v of the auxiliary data.

Image α is used as an input to the SRGAN resnet until the 8th residual block, at which point v is concatenated. The result is used as input to the second half of the residual blocks, and then into the SRGAN upsampling layer. MetaSRGAN is trained using the Adam optimizer, with a learning rate of 0.0002, and where $\beta = [0.5, 0.999]$.

MetaSRGAN uses the same discriminator and loss functions as SRGAN during training. Training was done with the full QuickRender dataset over a total of 200 epochs. This took approximately 48 hours on a single consumer-grade RTX 4090 GPU.

Model	SSIM	PSNR
SRGAN	0.6440	23.3231
MetaSRGAN	0.6958	26.4128

Table 1: SSIM and PSNR metrics showing improvement with MetaSRGAN as compared to SRGAN

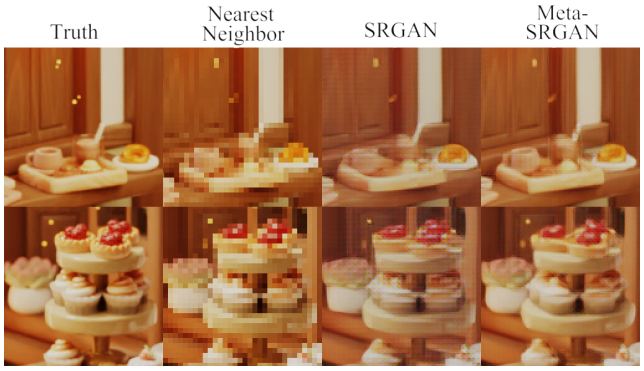

Figure 2: Result of SRGAN and MetaSRGAN when used on a scene outside of the dataset. Blender scene by Nicole Morena artstation.com/nickyblender

Results show that MetaSRGAN improves upon SRGAN in both ability to generalize as well as fine detail reconstruction as shown in figure 2.

When tested with inferring 2k and 4k square images from 512px and 1024px images respectively, MetaSRGAN achieved an average of 90.52% reduction in render times relative to rendering the full resolution output.

Future Work

To expand the applicability of QuickRender, we plan to render stereo images. Additionally, expanding the range of elements that are randomized with each procedural seed would allow for a larger dataset to be generated. In particular, traffic scenes could be an effective candidate, allowing for greater potential application in fields such as self driving vehicles. Finally, as SSIM and PSNR are not always indicative of improvement in super resolution tasks (Ledig et al. 2017), later versions could include other metrics such as mean opinion score.

Conclusion

The QuickRender dataset allows for a broad range of networks to utilize or generate auxiliary data relating to a base image. QuickRender maintains photo-realism while containing meaningfully diverse images to help with network generalization. Lastly, the dataset was demonstrated to be useful in the practical superresolution imaging task, yielding improvements in both image quality metrics and subjective assessment as compared to the original SRGAN.

Acknowledgments

This work was supported by the Faculty of Science and Technology Undergraduate Student Research Fund at Mount Royal University.

References

Bao, W.; Lai, W.-S.; Ma, C.; Zhang, X.; Gao, Z.; and Yang, M.-H. 2019. Depth-Aware Video Frame Interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3703–3712.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; van der Smagt, P.; Cremers, D.; and Brox, T. 2015. FlowNet: Learning Optical Flow With Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2758–2766.

Gaidon, A.; Wang, Q.; Cabon, Y.; and Vig, E. 2016. VirtualWorlds as Proxy for Multi-object Tracking Analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4340–4349. Las Vegas, NV, USA: IEEE. ISBN 978-1-4673-8851-1.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361.

Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; and Shi, W. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4681–4690.

Li, Z.; and Snavely, N. 2018. MegaDepth: Learning Single-View Depth Prediction From Internet Photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2041–2050.

Maximov, M.; Galim, K.; and Leal-Taixe, L. 2020. Focus on Defocus: Bridging the Synthetic to Real Domain Gap for Depth Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1071–1080.

Mayer, N.; Ilg, E.; Hausser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4040–4048.

Sun, D.; Vlasic, D.; Herrmann, C.; Jampani, V.; Krainin, M.; Chang, H.; Zabih, R.; Freeman, W. T.; and Liu, C. 2021. AutoFlow: Learning a Better Training Set for Optical Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10093–10102.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8934–8943.

Xue, T.; Chen, B.; Wu, J.; Wei, D.; and Freeman, W. T. 2019. Video Enhancement with Task-Oriented Flow. *International Journal of Computer Vision*, 127(8): 1106–1125.