Deferred Neural Rendering for View Extrapolation

Tobias Bertel
University of Bath

Yusuke Tomoto
Fyusion Inc.

Srinivas Rao
Fyusion Inc.

Rodrigo Ortiz-Cayon
Fyusion Inc.

Stefan Holzer
Fyusion Inc.

Christian Richardt
University of Bath

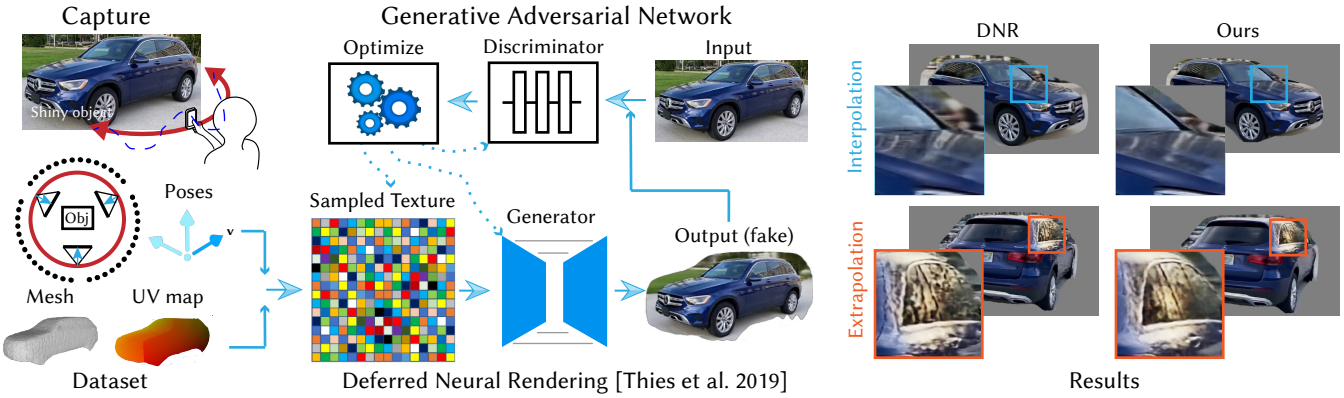

Figure 1: We capture an input video with a consumer camera, estimate camera poses, reconstruct a mesh and uv-map it. We extend Deferred Neural Rendering [Thies et al. 2019] (blue) to enable smooth extrapolation of novel viewpoints (orange).

ABSTRACT

Image-based rendering methods that support visually pleasing specular surface reflections require accurate surface geometry and a large number of input images. Recent advances in neural scene representations show excellent visual quality while requiring only imperfect mesh proxies or no surface-based proxies at all. While providing state-of-the-art visual quality, the inference time of learned models is usually too slow for interactive applications. While using a casually captured circular video sweep as input, we extend Deferred Neural Rendering to extrapolate smooth viewpoints around specular objects like a car.

CCS CONCEPTS

• **Computer graphics** → **Neural rendering**.

KEYWORDS

Novel-view synthesis, surface light field, extrapolation

ACM Reference Format:
Tobias Bertel, Yusuke Tomoto, Srinivas Rao, Rodrigo Ortiz-Cayon, Stefan Holzer, and Christian Richardt. 2020. Deferred Neural Rendering for View Extrapolation. In *SIGGRAPH Asia 2020 (SA '20 Posters), December 04-13, 2020.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3415264.3425441

SA '20 Posters, December 04-13, 2020, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8113-0/20/11.
https://doi.org/10.1145/3415264.3425441

1 INTRODUCTION

The reconstruction quality of casually captured data and the chosen scene representation constrain the development of interactive visual experiences, in particular in real-world environments. Traditionally, view synthesis of real-world environments is performed via image-based rendering (IBR), whose quality is mostly limited by the accuracy of the available scene *proxy* geometry. If RGB-*D* video is available, Park et al. [2020] show how to estimate scene properties like material and environment map, and use a physically motivated neural renderer for compositing Fresnel effects. We focus on *neural* scene representations and rendering methods that do not rely on explicit geometry at all, i.e. *posed* images only, or address the usage of an imperfect proxy geometry. *Volumetric* scene representations, e.g. NeRF [Mildenhall et al. 2020], register all input viewpoints within a learned semi-transparent volume. While the results are state-of-the-art visually, the inference time of the trained models is prohibitively slow for interactive applications. Liu et al. [2020] recently presented a sparse neural voxel representation suitable for indoor environments that renders 1–2 fps. *Image translation* methods used for image synthesis [Isola et al. 2017] show temporal artefacts since there is no global registration of the input views and they cannot be expected to work for non-diffuse scene objects. Hedman et al. [2018] use an *imperfect* proxy geometry and learn how to blend multiple images to mitigate rendering artefacts.

2 OUR APPROACH

The baseline of our system trains a generative adversarial network (GAN) to learn a mapping from view-dependent *neural features* to object appearance [Thies et al. 2019]. The learned features extracted from the neural texture are interpreted by the generator as a *learned* surface light field. We present a set of baseline extensions which lead to improved *extrapolation* performance. Note that the baseline,

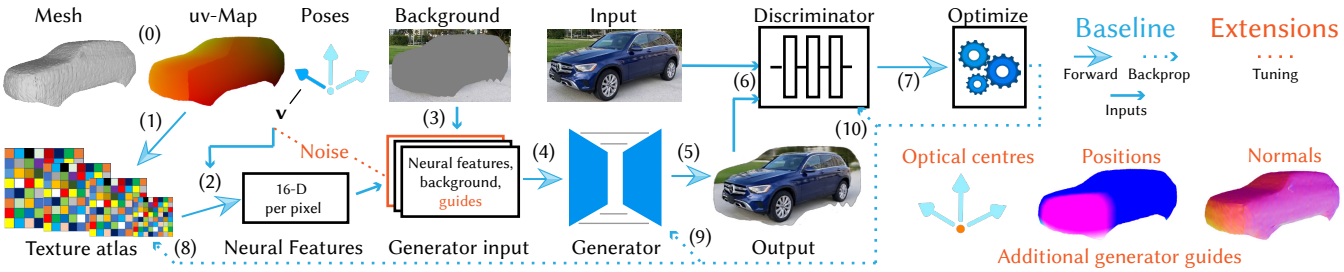

Figure 2: Overview of our baseline approach (blue) and our proposed extensions (orange). See detailed description in the text.

similar to most learning-based techniques, is designed to *interpolate* the training corpus. Deferred neural rendering is motivated by adding *learnable* components to the *deferred rendering* pipeline [Ritschel et al. 2012]. An overview of the baseline and our extensions is given in Figure 2. Mathematically, the goal is to find a combination of neural texture $\mathcal{T}$ and neural renderer $\mathcal{R}$ that minimizes the image re-rendering loss $\mathcal{L}$ over the training dataset $\mathcal{D}$ of M posed images $\mathcal{D} = \{I_:, p_:\}_{:=1}^{"}$ created from our capture. $I_:$ is the k-th image of the training dataset and its corresponding camera pose $p_:$, i.e. viewing direction v and optical centre c. The optimal neural texture $\mathcal{T}^*$ and renderer $\mathcal{R}^*$ are obtained by solving:

$$\mathcal{T}^*, \mathcal{R}^* = \arg\min_{\mathcal{T},\mathcal{R}} \tilde{O}_{3\in\mathcal{D}} \mathcal{L}(A(d) \mid F_3(\mathcal{T}), G_3(\mathcal{R})). \quad (1)$$

The baseline is obtained by setting $F_3(\cdot) = G_3(\cdot) = id$. The augmentation operator $A(\cdot)$ needs to return crops during training and its input during testing (i.e. identity). Our extensions address (i) augmenting training samples $d \in \mathcal{D}$, (ii) adding inputs to the renderer $\mathcal{R}$, and (iii) *injecting noise* into the *view*- and thus *dataset*-dependent feature generation $F(\cdot)$ or the *generator guides* $G(\cdot)$.

Forward pass. See (0–7) in Figure 2. Per data item $d \in \mathcal{D}$: (0) Rasterize a viewpoint and obtain uv-map (deferred rendering). (1) Use uv-map to look up texture atlas $\mathcal{A}$. (2) Retrieve neural features f, 16-D texels, from $\mathcal{A}$. The viewing direction $\mathsf{v} \in p \in d$ is converted to spherical harmonics. The first 2 bands are used, i.e. 9 coefficients, which are multiplied with the neural texel channels 4–12 of f_8. (3) Create background by eroding the uv-map from the training image I. (4) Encode generator input and decode it to produce output image (5). Note that the generator *interprets* the feature encoding used in (2). (6) Feed output and target into discriminator. (7) Optimize.

Backpropagation. See (8–10) in Figure 2. (8) Update atlas $\mathcal{A}$, keep view-dependent (specular) texels on finer levels. Regress neural feature channels $f[0:2]$ with the training image I. The motivation here is to get an estimate of the diffuse color of the surface. (9) Update generator via loss between the generated output O and data item image I. (10) Update discriminator.

Our extensions. See orange annotations in Figure 2. The baseline is extended in three ways: (i) A guided augmentation procedure $A(\cdot)$ is introduced that focuses on *poorly inferred image regions* when fetching dataset items d leading to more efficient training. SSIM is used to determine these image regions. (ii) Additional viewpoint information is added to the generator input, specifically optical centres, positions and normals. (iii) Noise is added to viewing directions v and newly added guidance signal of the generator input G_3.

Note that the noise injected into the viewing directions v reduces extrapolation artefacts significantly.

3 CONCLUSION

Individual viewpoints can be extrapolated smoothly, but there is no guarantee for *inter-frame coherency*. Note that the resulting *temporal flickering* could be reduced by: (i) providing a more accurate proxy geometry, (ii) increasing the image density and (iii) staying closer to the input viewpoints. While the baseline method is designed to *interpolate* the training corpus, as well as most other popular neural scene representations (e.g. NeRF), extrapolation causes uncertainty and thus noise. Our extensions technically trade this noise with blur. It seems promising to incorporate more object and scene information into the process and think of *inpainting* neural scene representations for view extrapolation tasks. It would be great to generalise the current representation in a physically motivated manner and surround the whole system with a differentiable rendering system to learn the components needed for real-time photo-real view synthesis. The current representation misses editing capabilities which hinders its practical applications.

ACKNOWLEDGMENTS

We want to thank Justus Thies for fruitful discussions and sharing the code for DNR. This work was supported by EU Horizon 2020 MSCA grant FIRE (665992), the EPSRC Centre for Doctoral Training in Digital Entertainment (EP/L016540/1), RCUK grant CAMERA (EP/M023281/1), an EPSRC-UKRI Innovation Fellowship (EP/S001050/1), and a Rabin Ezra Scholarship.

REFERENCES

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep Blending for Free-viewpoint Image-based Rendering. *ACM Trans. Graph.* 37, 6 (2018), 257:1–15. https://doi.org/10.1145/3272127.3275084

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*. https://doi.org/10.1109/CVPR.2017.632

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural Sparse Voxel Fields. arXiv:2007.11571

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*. https://doi.org/10.1007/978-3-030-58452-8_24

Jeong Joon Park, Aleksander Holynski, and Steve Seitz. 2020. Seeing the World in a Bag of Chips. In *CVPR*. https://doi.org/10.1109/CVPR42600.2020.00149

Tobias Ritschel, Carsten Dachsbacher, Thorsten Grosch, and Jan Kautz. 2012. The State of the Art in Interactive Global Illumination. *Comput. Graph. Forum* 31, 1 (Feb. 2012), 160–188. https://doi.org/10.1111/j.1467-8659.2012.02093.x

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred Neural Rendering: Image Synthesis Using Neural Textures. *ACM Trans. Graph.* 38, 4 (July 2019), 66:1–12. https://doi.org/10.1145/3306346.3323035