Pacific Graphics (2021)
M. Okabe, S. Lee, B. Wuensche, and S. Zollmann (Editors)

Short Paper

Neural Screen Space Rendering of Direct Illumination

Christian Suppan[1†], Andrew Chalmers[1], Junhong Zhao[1], Alex Doronin[2], Taehyun Rhee[1‡]

[1] Victoria University of Wellington, Computational Media Innovation Centre, New Zealand
[2] Victoria University of Wellington, School of Engineering and Computer Science, New Zealand

Abstract

Neural rendering is a class of methods that use deep learning to produce novel images of scenes from more limited information than traditional rendering methods. This is useful for information scarce applications like mixed reality or semantic photo synthesis but comes at the cost of control over the final appearance. We introduce the Neural Direct-illumination Renderer (NDR), a neural screen space renderer capable of rendering direct-illumination images of any geometry, with opaque materials, under distant illuminant. The NDR uses screen space buffers describing material, geometry, and illumination as inputs to provide direct control over the output. We introduce the use of intrinsic image decomposition to allow a Convolutional Neural Network (CNN) to learn a mapping from a large number of pixel buffers to rendered images. The NDR predicts shading maps, which are subsequently combined with albedo maps to create a rendered image. We show that the NDR produces plausible images that can be edited by modifying the input maps and marginally outperforms the state of the art while also providing more functionality.

CCS Concepts

• ***Computing methodologies*** → ***Rendering; Neural networks;*** *Supervised learning by regression;*

1. Introduction

Modern rendering methods produce high quality images from detailed and precise scene representations such as three dimensional (3D) geometry, textures, and light sources. These representations are expensive to capture and time consuming to construct manually. Neural rendering is a relatively new class of rendering methods that circumvent this cost by leveraging deep learning in combination with capture-friendly data sources, such as colour and depth images. Even without the explicit 3D scene understanding afforded by traditional scene representations, neural renderers are capable of, but not limited to, mixed reality rendering, semantic photo synthesis, novel view synthesis, relighting, performance reenactament, and volumetric rendering.

As a new field, neural rendering still has several open research challenges. The state-of-the-art review [TFT*20] identified controllability and generaliseability as key challenges. The majority of existing neural rendering methods are purely image based, allowing no simple method for user control over the renders. Also, many neural rendering methods are trained for one or only a few specific cases and the cost of retraining them makes them infeasible for many real world applications.

To address these challenges, we present the Neural Direct-illumination Renderer (NDR), a Convolutional Neural Network (CNN) that maps screen space buffers of geometry and material properties and Spherical Harmonics (SH) encoded illumination to rendered images. Screen space buffers are well known and controllable representations that are significantly easier to obtain than complete 3D scene descriptions. The required screen space buffers can be extracted from a single image using state-of-the-art inverse rendering methods and depth sensors. Since the NDR is conditioned on so many different inputs, a single trained instance is able to render direct-illumination images of any geometry, with any opaque material, under any distant illuminant.

To the best of our knowledge the paper "Deep Shading" [NAM*17] is the only directly related work. They describe a family of CNNs that add screen space rendering effects to direct illumination renders, as well as a CNN that predicts diffuse shading with fixed illumination from a normal buffer referred to as "Real Shading". This is the first example of a CNN generating an image from scratch from a screen space buffer. We expand on the seminal Real Shading method by introducing dependence on albedo, surface roughness, and illumination which enables rendering with control over those properties with a single trained instance of our method.

Our main contribution to achieve this is the introduction of intrinsic image decomposition to reduce the complexity of the mapping between buffers and rendered images that the CNN has to learn. Instead of the final render, the NDR outputs diffuse and spec-

† suppanchri@myvuw.ac.nz
‡ taehyun.rhee@ecs.vuw.ac.nz

DOI: 10.2312/pg.20211385

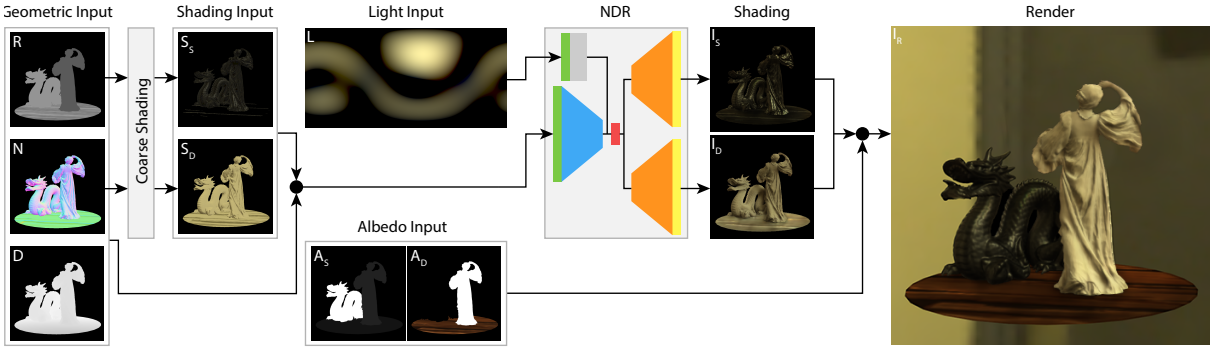

Figure 1: *Dataflow of the Neural Direct-Illumination Renderer showing how Light, Albedo, and Geometric inputs are processed to produce a final render. "Figure of a Dancer" by Agathon Léonard [Aga] is provided by Smithsonian 3D and licensed under CC0 1.0 [Creb].*

ular shading maps. These are then combined with albedo to create the final render with simple arithmetic operations. This effectively eliminates a known operation from the optimisation task. Reproduction of high frequency details in albedo is guaranteed and the learning of shading effects is stabilised by eliminating the need for high albedo to make them visible to the loss function.

2. Related Work

Deferred shading: Deferred shading [HH04] is a popular screen space rendering method. In its most basic form, deferred shading produces only direction dependant shading. Ambient occlusion and indirect illumination can be added through screen space methods [BS08, RGS09]. Shadows may also be added using shadow maps [SKVW*92] or shadow volumes [Cro77], but this requires fully defined geometry and accurately defined illumination.

Neural Rendering: Many neural image synthesis methods have been variably classified as novel view synthesis or image based rendering approaches. These are methods that produce new images of a scene based on an existing collection of observations. Kulkarni et al. [KWKT15] encode multiple observations as a disentangled latent space that can be substituted piecewise to enact camera transformations in the output. Sitzmann et al. [STH*19] and Nguyen et al. [NPLT*19] use 3D convolutions to allow transformations to be applied directly to the latent space. Aliev et al. [AUL19] project observations into point clouds of neural features that can be reprojected before decoding to create novel views. Meshry et al. [MGK*19] use point clouds and additionally handle varying lighting conditions and partial occlusions in input images. Thies et al. [TZN19] and Chen et al. [CCZ*20] encode multiple observations into neural textures that can be transformed to create novel views and geometric distortions. Lombadri et al. [LSSS18,LSS*19] reconstruct faces from few precisely calibrated camera views of the face. Unlike image based rendering, the input buffers of our method can be edited directly to enact a change in shape, material, or illumination.

CNNs have been applied to image relighting. Meka et al. [MPH*20] encode light stage observations into neural textures to achieve free viewpoint relighting. Xu et al. [XSHR18] learn optimal samples to relight images with accurate view dependant effects. Sun et al. [SBT*19] achieve single image relighting for portrait images.

There is a limited body of existing work that mimics the behaviour of traditional renderers, mapping traditional scene descriptions to images. Nguyen et al. [NPLBY18] use a novel projection unit to map a voxel grid to diffuse shading. Rematas et al. [RF19] map a voxel grid to images featuring cast shadows, reflections, and spatially varying materials. This impressive result is based on fully defined scene geometry under point lighting. Our method aims to function with partial geometry under area lighting. Li et al. [LXR*18] render global illumination by using stacked networks to predict subsequent bounces of light from direct illumination images and deferred buffers. Nalbach et al. [NAM*17] describe a family of CNNs that add screen space rendering effects to direct illumination renders as well as Real Shading, a CNN that maps screen space normals to diffuse shading. Our method builds on this work to handle specular materials and variable illumination at inference time.

3. Method

3.1. Network Structure

The aim of our method is to render direct illumination images, including occlusion based effects, from single viewpoint screen space information. Inspired by the success of deep learning methods at various image generation tasks, we propose the Neural Direct-illumination Renderer (NDR), a novel CNN for mapping these unique inputs to shading maps. Formally, given normals N, roughness R, a depth image D, approximate diffuse S_D and specular S_S shading, and SH coefficients for illumination L the NDR, denoted as $r(.)$, estimates specular shading map I_S and diffuse shading map I_D:

$$NDR: \quad r(\{N,R,D,L,S_D,S_S\}) \rightarrow \{I_S,I_D\} \tag{1}$$

An image is rendered from these shading maps by multiplying them with their respective albedos, as per the intrinsic image model [BTHR78]. Formally:

$$I_R = I_S * A_S + I_D * A_D \tag{2}$$

where I_R is the rendered image and A_S, and A_D are the specular and

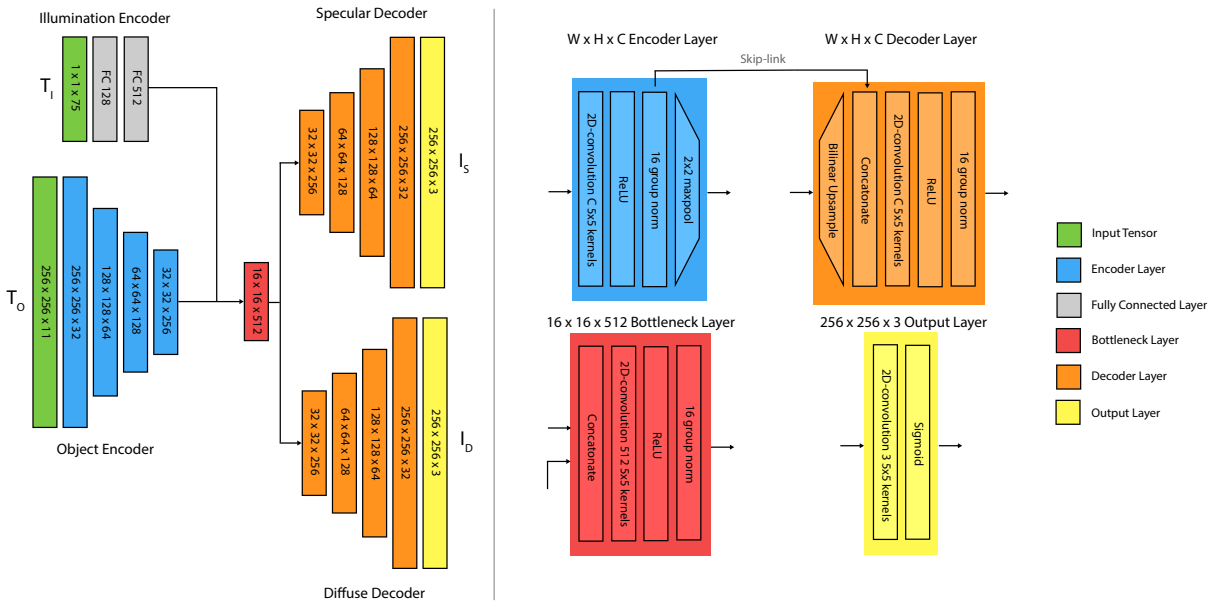

Figure 2: *Detailed network structure of the Neural Direct-Illumination Renderer. The W, H values given are those used during our experiments. The W, H values could be varied arbitrarily without modification of the network structure as it is fully convolutional.*

diffuse albedo maps respectively. The dataflow of our rendering pipeline is illustrated in Figure 1.

The coarse shading maps S_D and S_S are generated by considering each pixel a lone point in space, ignoring all occlusions, because the full 3D context is not available. S_D and S_S are calculated by importance sampling the the lambertian and GGX BRDFs with 64 samples per pixel. A low sample count and a 32x16px radiance map are used to ensure the execution is fast enough to train with and to simulate that detailed illumination may not be available in single view point input scenarios. We similarly justify the choice of spherical harmonics.

The NDR is a U-Net [RFB15] with two modifications; a second decoder, and injection of illumination information into the bottleneck. The second decoder is just a parallel copy of the normal U-net decoder that shares the same bottleneck and skip link inputs. Illumination is encoded as a vector of 75 SH coefficients. The entirety of the illumination information must be available to every pixel in the output. This is done by passing the vector through two fully connected layers and depth-wise concatenating the resultant feature vector to every spatial position of the bottleneck input. We define the tensor passed to the normal U-net encoder as $T_O : \{N, R, D, S_D, S_S\}$ and the tensor passed to the illumination encoder as $T_I : \{L\}$. The detailed network architecture is illustrated in Figure 2. Note that the output layers feature a sigmoid activation. The inverse tone mapping operator $-I/(I-1)$ is used to map the outputs to HDR shading maps.

3.2. Dataset

To train the NDR we require a dataset of direct illumination renders, shading maps, and parameter buffers of objects to render. We procedurally generate these objects by combining height mapped primitive shapes in the manner of Xu et al. [XSHR18] and Li et al. [LXR*18]. The objects are textured with random crops taken from the texture dataset provided by Deschaintre et al. [DAD*18]. The objects are illuminated by radiance maps taken from the Laval Indoor dataset [GSY*17]. We render our datasets with PBRT-v3 [PJH16] with an implementation of the Cook-Torrance model with GGX distributions described by Karis [KG13]. We use masks for loss calculations, which are generated by converting depth maps to binary masks and applying morphological erosion. We generate 60,000 training samples and multiple sets of 1,200 testing samples.

3.3. Training

The NDR is trained end-to-end in a supervised manner with the loss function $\mathcal{L} = \mathcal{L}_d + \mathcal{L}_s + \mathcal{L}_r$, where $\mathcal{L}_d$, $\mathcal{L}_s$, and $\mathcal{L}_r$ are the Structural Dissimilarity (DSSIM) [WBSS04] loss on the diffuse shading, specular shading, and final rendered image. The shading maps are separately supervised to ensure that strong gradients are provided to each decoder during each training step, irrespective of how little that shading component contributes to the final image. To stabilise training, all inputs and targets with unbounded value ranges (depth and all shading maps) are remapped to the range [0,1] with Reinhard tone mapping [RSSF02]. We train the network for 40 epochs using the Adam optimiser [KB14] with a fixed learning rate of 1e-4 and batch size of 16 using an NVIDIA Quadro RTX 6000. This took approximately 15 hours.

Once trained, evaluating the entire pipeline takes approximately 0.046 seconds with, and 0.006 seconds without including the time to generate the coarse shading maps. The fast execution and trivial differentiability of the NDR may be useful in applications such as interactive mixed reality and render loss driven optimisations.

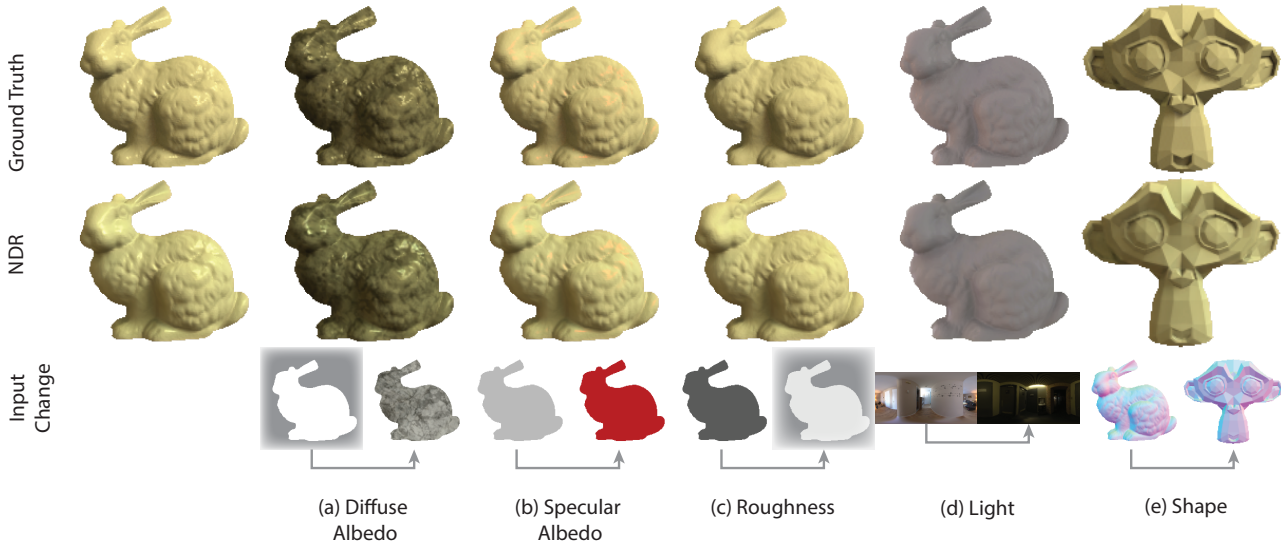

Figure 3: *Comparison of images rendered by the NDR to the path traced ground truth when changing material (a-c), illumination (d), and geometry (e). The changes in input are shown below each example.*

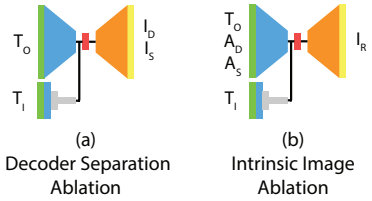

Figure 4: *Abridged network structures of ablated NDR variants.*

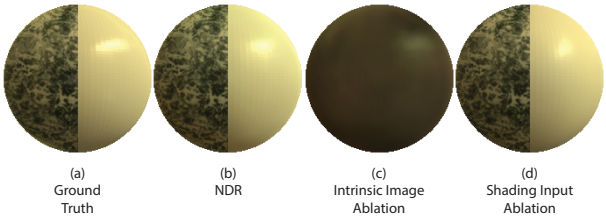

Figure 5: *Comparison of images rendered by the NDR and its ablated variants without intrinsic image decomposition and shading input.*

4. Experiments

In this section we demonstrate the efficacy of the NDR as a renderer, support the choice of intrinsic image decomposition with an ablation study, and evaluate it against the preceding Real Shading.

4.1. Functionality

As a renderer the NDR must be able to evaluate the interaction of light, material and geometry. We demonstrate this with a series of qualitative examples of how the NDR responds to changes in input in Figure 3. Note the high frequency albedo, sharp specular highlights, and plausible self occlusion.

Table 1: *Mean L2 and DSSIM errors of the NDR and its ablated variants.*

	L2	DSSIM
NDR	**0.00017**	**0.0031**
Decoder Sep. Abl.	0.00017	0.0031
Intrinsic Img. Abl.	0.00087	0.0970
Shading Input Abl.	0.00017	0.0035
SH Input Abl.	0.00018	0.0045
Depth Input Abl.	0.00017	0.0037

4.2. Ablation

We conduct a structural ablation to show the efficacy of our main contribution, the use of an intrinsic image formulation. We first ablate the separation of the decoders, replacing them with a single decoder with twice the number of output channels per level. We then ablate the intrinsic image formulation by additionally appending albedos to T_o, targeting I_R directly. These ablated structures are shown in Figure 4. Quantitative results are given in Table 1. Ablating the decoder separation causes no significant degradation but increases the number of weights by 9%. This shows that the shading components can be reasoned about separately and enforcing this in the network structure improves the representational efficiency. Ablating the intrinsic image formulation causes the NDR to not converge. Albedo causes a large amount of the variance in target images, and the learning of a near identity transform is required to create a good enough baseline to stably optimise for shading effects. However, without significant modifications to the training procedure shading effects are significant enough to interfere with the learning of said near identity transform. Therefore, the ablated

Figure 6: *Comparison of images rendered by the NDR and comparable methods under fixed illumination. The same object is shown with a material that is purely diffuse, purely specular, and a mixture of both. The model "Decorative Vase" by fresherator2 [fre20] is licensed under CC BY 4.0 [Crea].*

Table 2: *Mean L2 and DSSIM errors of the NDR and comparable methods over our two test datasets.*

	Diffuse Material Fixed Illumination		Complex Material Varying Illumination	
	L2	DSSIM	L2	DSSIM
NDR	0.00029	**0.0109**	**0.00017**	**0.0031**
Real Shading	0.00093	0.0231	N/A	N/A
pix2pix	1.22490	0.4051	0.00045	0.0200
Shading Input	0.00171	0.0814	0.00032	0.0240
Projection Meshing	**0.00024**	0.0253	0.00018	0.0189

NDR achieves only vague indications of shading and loses structure due to albedo, see Figure 5.

We additionally conduct a data ablation to demonstrate that potentially redundant inputs provide useful information. Shading and SH inputs both encode directional illumination, but removing either reduces numerical accuracy by 12% and 45% respectively. While it has a smaller numerical effect, the shading input is key for creating correctly shaped highlights, see Figure 5. Normal and depth inputs are theoretically redundant as they are related by an integral. Normal input is required to generate the shading input and cannot be ablated. Ablating depth input reduces numerical accuracy by 19%, likely due to reduced ability to reason about occlusions.

4.3. Evaluation

Given that we are expanding the functionality of the first example of a CNN based screen space rendering pipeline there is no direct point of comparison. We compare to Real Shading [NAM*17] itself to verify that the increased input space has not degraded performance, pix2pix [IZZE17] to show the benefits of our specialised architecture over a generic one, the shading input to show the NDR is not just an expensive identity mapping, and path traced renders of meshes constructed by projecting screen space depth maps [PCR18]. All methods receive identical inputs of 256x256px buffers and 32x16px/75 SH coefficients. Our light encoder was added to pix2pix to allow for variable illumination experiments. We compare over two datasets. One consists of untextured diffuse objects suitable for evaluating Real Shading. The other consist of textured objects under variable illumination.

Quantitative results are given in Table 2. A qualitative example is given in Figure 6. The NDR numerically outperforms Real Shading, even with a larger input space. Both produce visually plausible renders but Real Shading does not account for specular materials. pix2pix has an order of magnitude greater DSSIM error than the NDR due to its lack of task specific design. Without intrinsic image decomposition the diffuse and specular contributions become entangled, e.g. the exemplar vase rendered with both specular and diffuse contributions is significantly brighter than the sum of the separately predicted components. As expected, the shading input performs poorly due to its lack of occlusions and limited quality inputs. The low resolution radiance map and lack of smoothing from considering each surface point in isolation results in excessively sharp discontinuities in shading. The NDR significantly improves on its input. Generating a mesh by projecting the vertices of a plane mesh along view vectors based on a depth map has many obvious failure cases where depth discontinuities are erroneously connected or object edges being projected to infinity causes excess occlusions. These failure cases are significant enough that the NDR numerically outperforms this method, even though the expensive process of path tracing provides very accurate evaluation of unoccluded points. Such a meshing method is also vulnerable to noise, e.g. the scan of the exemplar vase features just one tiny depth map error that still causes a noticeable artefact. Their layered spatial filtering operations make CNNs, such as the NDR, resilient to such noise.

5. Conclusion

We present the Neural Direct-illumination Renderer, a method for rendering images of arbitrary objects under distant illumination from screen space buffers. The NDR offers a controllable and generalised alternative to traditional rendering approaches when only partial scene descriptions of limited quality are available. The NDR currently cannot evaluate multiple light-material interactions but auxiliary methods such as GINet [LXR*18] may be appended to add such functionality. By leveraging intrinsic image image formation our novel Convolutional Neural Network architecture outperforms and offers a greater range of functionality than the current state of the art. The render quality of the NDR may be further improved in future with access to real training data.

6. Acknowledgments

This project was funded by the Smart Ideas Endeavour Fund from MBIE and in part by the Entrepreneurial University Programme from TEC in New Zealand.

References

[Aga] AGATHON LÉONARD: "figure of a dancer" accessed jun., 2020. Smithsonian 3D. URL: https:

//3d.si.edu/object/3d/figure-dancer:88de08dd-b8ab-470a-b987-ed6fe35def04. 2

[AUL19] ALIEV K.-A., ULYANOV D., LEMPITSKY V.: Neural point-based graphics. *arXiv preprint arXiv:1906.08240* (2019). 2

[BS08] BAVOIL L., SAINZ M.: Screen space ambient occlusion. *NVIDIA developer information: http://developers. nvidia. com 6* (2008). 2

[BTHR78] BARROW H., TENENBAUM J., HANSON A., RISEMAN E.: Recovering intrinsic scene characteristics. *Comput. Vis. Syst 2*, 3-26 (1978), 2. 2

[CCZ*20] CHEN Z., CHEN A., ZHANG G., WANG C., JI Y., KUTULAKOS K. N., YU J.: A neural rendering framework for free-viewpoint relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 5599–5610. 2

[Crea] CREATIVE COMMONS: "creative commons legal code cc by 4.0" accessed dec., 2020. URL: https://creativecommons.org/licenses/by/4.0/legalcode. 5

[Creb] CREATIVE COMMONS: "creative commons legal code cc0 1.0" accessed dec., 2020. URL: https://creativecommons.org/publicdomain/zero/1.0/legalcode. 2

[Cro77] CROW F. C.: Shadow algorithms for computer graphics. *Acm siggraph computer graphics 11*, 2 (1977), 242–248. 2

[DAD*18] DESCHAINTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (ToG) 37*, 4 (2018), 1–15. 3

[fre20] FRESHERATOR2: Decorative vase. Sketchfab, 2020. URL: https://sketchfab.com/3d-models/decorative-vase-fd455b8cb6da4cb0b4548a0f2e74ddfe. 5

[GSY*17] GARDNER M.-A., SUNKAVALLI K., YUMER E., SHEN X., GAMBARETTO E., GAGNÉ C., LALONDE J.-F.: Learning to predict indoor illumination from a single image. *arXiv preprint arXiv:1704.00090* (2017). 3

[HH04] HARGREAVES S., HARRIS M.: Deferred shading. In *Game Developers Conference* (2004), vol. 2, p. 31. 2

[IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 1125–1134. 5

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 3

[KG13] KARIS B., GAMES E.: Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice 4* (2013). 3

[KWKT15] KULKARNI T. D., WHITNEY W. F., KOHLI P., TENENBAUM J.: Deep convolutional inverse graphics network. In *Advances in neural information processing systems* (2015), pp. 2539–2547. 2

[LSS*19] LOMBARDI S., SIMON T., SARAGIH J., SCHWARTZ G., LEHRMANN A., SHEIKH Y.: Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 65. 2

[LSSS18] LOMBARDI S., SARAGIH J., SIMON T., SHEIKH Y.: Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–13. 2

[LXR*18] LI Z., XU Z., RAMAMOORTHI R., SUNKAVALLI K., CHANDRAKER M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–11. 2, 3, 5

[MGK*19] MESHRY M., GOLDMAN D. B., KHAMIS S., HOPPE H., PANDEY R., SNAVELY N., MARTIN-BRUALLA R.: Neural rerendering in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 6878–6887. 2

[MPH*20] MEKA A., PANDEY R., HÄNE C., ORTS-ESCOLANO S., BARNUM P., DAVID-SON P., ERICKSON D., ZHANG Y., TAYLOR J., BOUAZIZ S., ET AL.: Deep relightable textures: volumetric performance capture with neural rendering. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–21. 2

[NAM*17] NALBACH O., ARABADZHIYSKA E., MEHTA D., SEIDEL H.-P., RITSCHEL T.: Deep shading: convolutional neural networks for screen space shading. In *Computer graphics forum* (2017), vol. 36, Wiley Online Library, pp. 65–78. 1, 2, 5

[NPLBY18] NGUYEN-PHUOC T. H., LI C., BALABAN S., YANG Y.: Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. In *Advances in Neural Information Processing Systems* (2018), pp. 7891–7901. 2

[NPLT*19] NGUYEN-PHUOC T., LI C., THEIS L., RICHARDT C., YANG Y.-L.: Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 7588–7597. 2

[PCR18] PETIKAM L., CHALMERS A., RHEE T.: Visual perception of real world depth map resolution for mixed reality rendering. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2018), pp. 401–408. 5

[PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016. 3

[RF19] REMATAS K., FERRARI V.: Neural voxel renderer: Learning an accurate and controllable rendering tool. *arXiv preprint arXiv:1912.04591* (2019). 2

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (2015), Springer, pp. 234–241. 3

[RGS09] RITSCHEL T., GROSCH T., SEIDEL H.-P.: Approximating dynamic global illumination in image space. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (2009), pp. 75–82. 2

[RSSF02] REINHARD E., STARK M., SHIRLEY P., FERWERDA J.: Photographic tone reproduction for digital images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), pp. 267–276. 3

[SBT*19] SUN T., BARRON J. T., TSAI Y.-T., XU Z., YU X., FYFFE G., RHEMANN C., BUSCH J., DEBEVEC P., RAMAMOORTHI R.: Single image portrait relighting. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* (2019). 2

[SKVW*92] SEGAL M., KOROBKIN C., VAN WIDENFELT R., FORAN J., HAEBERLI P.: Fast shadows and lighting effects using texture mapping. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (1992), pp. 249–252. 2

[STH*19] SITZMANN V., THIES J., HEIDE F., NIESSNER M., WETZSTEIN G., ZOLLHOFER M.: Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 2437–2446. 2

[TFT*20] TEWARI A., FRIED O., THIES J., SITZMANN V., LOMBARDI S., SUNKAVALLI K., MARTIN-BRUALLA R., SIMON T., SARAGIH J., NIESSNER M., ET AL.: State of the art on neural rendering. *arXiv preprint arXiv:2004.03805* (2020). 1

[TZN19] THIES J., ZOLLHÖFER M., NIESSNER M.: Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–12. 2

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing 13*, 4 (2004), 600–612. 3

[XSHR18] XU Z., SUNKAVALLI K., HADAP S., RAMAMOORTHI R.: Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–13. 2, 3