

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Toward Relighting and View Synthesis for Human Portraits

Permalink

<https://escholarship.org/uc/item/26t98108>

Author

Sun, Tiancheng

Publication Date

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Toward Relighting and View Synthesis for Human Portraits

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Tiancheng Sun

Committee in charge:

Professor Ravi Ramamoorthi, Chair
Professor Manmohan Chandraker
Professor David Kriegman
Professor Hao Su
Professor Zhuowen Tu

2021

Copyright

Tiancheng Sun, 2021

All rights reserved.

The Dissertation of Tiancheng Sun is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

To my family.

TABLE OF CONTENTS

Dissertation Approval Page	iii
Dedication	iv
Table of Contents	v
List of Figures	vii
List of Tables	ix
Acknowledgements	x
Vita	xii
Abstract of the Dissertation	xiv
Chapter 1 Introduction	1
Chapter 2 Single Image Portrait Relighting	5
2.1 Introduction	5
2.2 Related Work	8
2.3 Learning Relighting for Portraits	12
2.3.1 Data	13
2.3.2 Network Architecture	16
2.3.3 Loss Function and Training	18
2.4 Evaluation	19
2.4.1 Relit Images	20
2.4.2 Estimated Lighting	26
2.5 Real-world Results	28
2.6 Conclusion	32
Chapter 3 Light Stage Super-Resolution: Continuous High-Frequency Relighting ...	34
3.1 Introduction	34
3.2 Related Work	37
3.3 Model	39
3.3.1 Active Set Selection	41
3.3.2 Alias-Free Pooling	44
3.3.3 Network Architecture	45
3.3.4 Loss Functions and Training Strategy	46
3.4 Evaluation	47
3.4.1 Ablation Study	49
3.4.2 Related Work Comparison	52
3.4.3 Light Stage Subsampling	54

3.5	Continuous High-Frequency Relighting	55
3.6	Conclusions and Future Work	60
Chapter 4	NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting	63
4.1	Introduction	63
4.2	Related Work	66
4.2.1	Portrait Appearance	66
4.2.2	Relighting	66
4.2.3	View Synthesis	67
4.2.4	Neural Rendering	67
4.3	Method	68
4.3.1	Light Transport	69
4.3.2	Neural Light-transport Field	71
4.3.3	Efficient Volume Rendering	74
4.3.4	Domain Adaptation	75
4.3.5	Implementation Detail	76
4.4	Results	78
4.4.1	Data	78
4.4.2	Comparison with previous methods	79
4.4.3	Ablations	81
4.4.4	Real Portraits	82
4.4.5	Limitations	84
4.5	Conclusions and Future Work	84
Chapter 5	Conclusion	86
	Bibliography	89

LIST OF FIGURES

Figure 1.1.	Toward Relighting and View Synthesis for Human Portraits	1
Figure 1.2.	Portrait Light on Google Pixel phones	3
Figure 2.1.	Single image portrait relighting	6
Figure 2.2.	System architecture of portrait relighting network	7
Figure 2.3.	Portraits captured from the light stage and relit under natural lighting	14
Figure 2.4.	Network architecture of portrait relighting network	16
Figure 2.5.	Comparison with previous works on portrait relighting	23
Figure 2.6.	Comparison with previous works on lighting transfer	24
Figure 2.7.	Ablation results on self-supervision	26
Figure 2.8.	Comparison with previous works on lighting estimation	27
Figure 2.9.	Relighting results on portraits captured from cellphones	28
Figure 2.10.	Relighting results of portraits with unexpected objects	30
Figure 2.11.	Illumination retargeting results on portraits taken from cellphones	30
Figure 2.12.	Lighting transfer results on portraits taken from cellphones	31
Figure 2.13.	Failure cases of the portrait relighting network	32
Figure 2.14.	Google Portrait Light on Pixel phones	33
Figure 3.1.	Light stage super-resolution	35
Figure 3.2.	Model architecture for light stage super resolution	40
Figure 3.3.	Active set selection	42
Figure 3.4.	Alias-free pooling	43
Figure 3.5.	Shadow movement visualization of different model	50
Figure 3.6.	Comparison with previous works on light stage super-resolution	51
Figure 3.7.	Ablation results with different numbers of lights on the light stage	54

Figure 3.8.	Plot of image quality respect to the number of lights on the light stage	55
Figure 3.9.	Continuous lighting interpolation between two point lights	56
Figure 3.10.	Environmental relighting with and without super-resolution	57
Figure 3.11.	Frequency analysis on light stage super-resolution	58
Figure 3.12.	Control shadow softness with and without super-resolution	60
Figure 4.1.	Neural light-transport field for portrait view synthesis and relighting	64
Figure 4.2.	Overview of neural light-transport field	69
Figure 4.3.	Detailed network structures of neural light-transport field	71
Figure 4.4.	Comparison with previous works on portrait view synthesis and relighting	79
Figure 4.5.	Ablation results on view synthesis and relighting	81
Figure 4.6.	View synthesis and relighting results with different number of portraits as the inputs	82
Figure 4.7.	Comparison with previous work on real captured portraits	83

LIST OF TABLES

Table 2.1.	Quantitative results on single image portrait relighting	21
Table 2.2.	Quantitative results on lighting transfer	24
Table 2.3.	Quantitative results on lighting estimation	27
Table 3.1.	Quantitative comparison on light stage super-resolution	49
Table 4.1.	Quantitative comparison on view synthesis and relighting	78
Table 4.2.	Quantitative comparison on the task of relighting and view synthesis indi- vidually	80
Table 4.3.	Quantitative comparison on different input view counts	82

ACKNOWLEDGEMENTS

I would like to give my greatest gratitude to my PhD advisor, Professor Ravi Ramamoorthi. I can never imagine a better PhD advisor than Professor Ramamoorthi: He is knowledgeable, responsible, and also very supportive of my personal choices throughout my PhD life. He gives me the freedom to work on the projects I am interested in, and provides valuable and timely guidance when I encountered obstacles. Most importantly, he teaches me how to do research, how to do research well, and I hope I can be a researcher as good as him in the future. I also want to thank Professor Manmohan Chandraker, Professor Hao Su, Professor David Kriegman, and Professor Zhuowen Tu for serving on my thesis committee and for providing their generous help and guidance.

The work is inseparable from the help of my amazing collaborators. Thanks to Professor Henrik Wann Jensen, Zexiang Xu, Sai Bi, Kai-En Lin, Alexandr Kuznetsov, Shilin Zhu for their generous help and valuable contribution to the projects. Thanks to Zhengqin Li, Lifan Wu, Jiyang Yu, Mohammad Sina Nabizadeh, Ling-Qi Yan for their constructive advice and fruitful discussions. Thanks to all my labmates and friends in UCSD for making my PhD life in San Diego enjoyable and delightful. I am fortunate enough to intern at Google twice and Facebook once. I would like to thank all my friends I met in Google: Jonathan T. Barron, Yun-Ta Tsai, Xiuming Zhang, Xuaner Zhang, Xueming Yu, Paul E. Debevec, Graham Fyffe, Jay Busch, Wan-Chun Ma, Peter Denny, Christoph Rhemann, and Sean Fanello. The success of portrait relighting cannot be achieved without your generous support on the project during and after my internship. My gratitude also goes generally to Google, which has generously supported me through Google PhD Fellowship. I also like to thank Giljoo Nam, Carlos Aliaga, Christophe Hery, Olivier Maury from Facebook, who made my internship in Facebook productive and enjoyable, even during the time of the pandemic.

I owe thanks to a very special person, my wife Meng Deng, for her continued and unfailing love, support, and understanding when I was pursuing my PhD degree. She was by my side every time I worked on a paper deadline. She would be excited for me even when I

achieved a small goal, and encourage me and back me up when my project was not going so well. Besides her generous support to my career, she has also made my everyday life full of love and happiness. Along with her, I also want to thank my parents, grandparents, and all my family for their continuous and unparalleled love, help, and support. I am super grateful to my parents for giving birth to me, and for providing me the opportunities and experiences which make me who I am.

This work is funded by a Jacobs Fellowship, the Ronald L. Graham Chair, a Google Fellowship, a Qualcomm FMA Fellowship, an Amazon research award, gifts from Adobe, and the UC San Diego Center for Visual Computing. This work is also supported by NSF grants 1617234, 1703957, and 1730158, ONR grants N000141712687, N000142012529, and N000141912293.

Chapter 2 is a reformatted version of the material as it appears in “Single Image Portrait Relighting”, Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E. Debevec, and Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 38 (4), 2019 [109]. The dissertation author was the primary investigator and author of this paper.

Chapter 3 is a reformatted version of the material as it appears in “Light stage super-resolution: continuous high-frequency relighting”, Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T. Barron, and Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 39 (6), 2019 [111]. The dissertation author was the primary investigator and author of this paper.

Chapter 4 is a reformatted version of the material as it appears in “NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting”, Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, and Ravi Ramamoorthi, Eurographics Symposium on Rendering, 2021 [110]. The dissertation author was the co-primary investigator and author of this paper.

VITA

2017-2021 Doctor of Philosophy, Computer Science, University of California, San Diego
2013-2017 Bachelor of Engineering, Computer Science, Tsinghua University

PUBLICATIONS

Shilin Zhu, Zexiang Xu, **Tiancheng Sun**, Alexandr Kuznetsov, Mark Meyer, Henrik Wann Jensen, Hao Su, Ravi Ramamoorthi. “Hierarchical Neural Reconstruction for Path Guiding Using Hybrid Path and Photon Samples,” ACM Transactions on Graphics (TOG) 40 (4), 2021

Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, Ravi Ramamoorthi. “NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting,” Eurographics Symposium on Rendering, 2021

Tiancheng Sun, Giljoo Nam, Carlos Aliaga, Christophe Hery, Ravi Ramamoorthi. “Human Hair Inverse Rendering using Multi-View Photometric data,” Eurographics Symposium on Rendering, 2021

Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, Ravi Ramamoorthi. “NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting,” Eurographics Symposium on Rendering, 2021

Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, **Tiancheng Sun**, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, Jonathan T Barron, Ravi Ramamoorthi, William T Freeman. “Neural light transport for relighting and view synthesis,” ACM Transactions on Graphics (TOG) 40 (1), 2021

Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T Barron, Ravi Ramamoorthi. “Light stage super-resolution: continuous high-frequency relighting,” ACM Transactions on Graphics (TOG) 39 (6), 2020

Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, Ravi Ramamoorthi. “Single image portrait relighting,” ACM Transactions on Graphics (TOG) 38 (4), 2020

Tiancheng Sun, Henrik Wann Jensen, Ravi Ramamoorthi. “Connecting measured BRDFs to analytic BRDFs by data-driven diffuse-specular separation,” ACM Transactions on Graphics (TOG) 37 (6), 2019

Tiancheng Sun, Yifan Peng, Wolfgang Heidrich. “Revisiting cross-channel information transfer for chromatic aberration correction,” IEEE International Conference on Computer Vision (ICCV), 2017

Tiancheng Sun, Yulong Wang, Jian Yang, Xiaolin Hu. “Convolution Neural Networks With Two Pathways for Image Style Recognition,” IEEE Transactions on Image Processing 26 (9), 2017

Tiancheng Sun, Ana Serrano, Diego Gutierrez, Belen Masia. “Attribute-preserving gamut mapping of measured BRDFs,” Eurographics Symposium on Rendering, 2017

ABSTRACT OF THE DISSERTATION

Toward Relighting and View Synthesis for Human Portraits

by

Tiancheng Sun

Doctor of Philosophy in Computer Science

University of California San Diego, 2021

Professor Ravi Ramamoorthi, Chair

Human portraits are ubiquitous in our everyday life. However, after we take the portraits using the camera, the lighting condition of the captured image cannot be easily changed. Previous works have investigated how to change the lighting condition of a portrait, but they usually require special hardware to achieve relighting. In this thesis, we study the problem of portrait relighting, aiming at accuracy and practicality. We present several novel approaches that enable relighting portraits after the capture.

With the goal of practical relighting that can be used in everyday life, we first develop *single image portrait relighting*. We leverage the prior of human portraits, and train a neural network that learns to perform relighting given only a single portrait. Our network can achieve

convincing relighting results with very limited computation power, enabling real-time portrait relighting applications.

Next, we focus on how to achieve precise relighting under point lights. We capture the portrait using discrete point lights under a light stage, and super-sample the location of the point lights to achieve high-frequency relighting. This enables finer controls on the shadows on human faces.

Finally, we investigate changing both the viewpoint and the lighting condition of captured portraits. To this end, we propose *neural light-transport field (NeLF)*. Given only 5 frontal portraits, this neural rendering approach can generate portraits from novel views under arbitrary natural lightings.

Chapter 1

Introduction

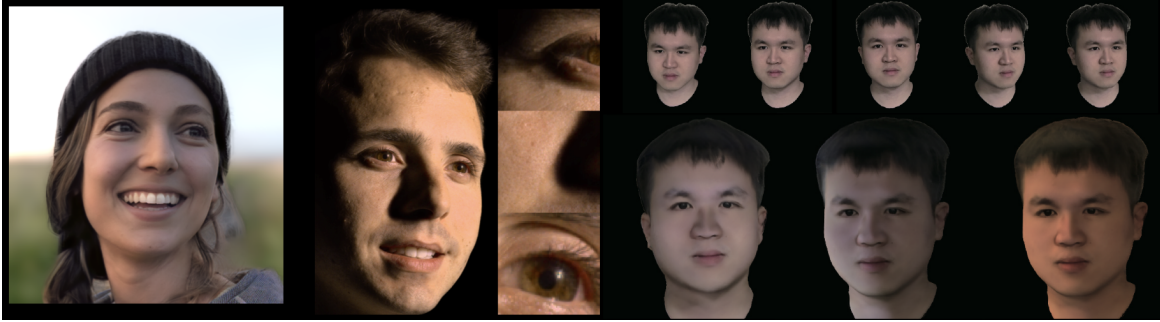


Figure 1.1. Toward Relighting and View Synthesis for Human Portraits

Lighting plays an important role in conveying the essence and the depth of the subject in a portrait photograph. Professional photographers will carefully control the lighting in their studio to manipulate the appearance of their subject, while amateur photographers are usually constrained to the illumination of their environment. Over the past decade, there has been some progress on technology that enables individuals to change the lighting condition of a portrait after the capture process. This kind of technique is called *relighting*.

Relighting is a problem that spans multiple domains in computer graphics. Given a scene, say a human face, in order to change the lighting, we first need to extract the information of the portrait from the captured images, which is usually called an *inverse rendering* process. Then, we need a proper *appearance model* for the portrait, in order to change the lighting information in an intuitive way. Finally, we need to render a new portrait under the new lighting, which usually involves some *forward rendering* techniques.

Plenty of prior works have explored techniques for relighting an image. However, their utility is usually limited due to requirements of specialized hardware, multiple images of the subject under controlled or known illuminations, or accurate models of geometry and reflectance. In this thesis, we aim to achieve efficient and practical relighting on human portraits taken under casual setups. In Chapter 2, we present an efficient and practical method to relight the portrait under arbitrary natural illuminations with only a single image. In Chapter 3, we investigate the problem of high-frequency relighting, enabling finer shadow control on portraits taken under a light stage. And in Chapter 4, we combine our relighting algorithm with view synthesis, and develop a novel approach that can achieve simultaneous view synthesis and relighting given 5 frontal portrait.

People enjoy taking portraits using their cellphones, but the lighting condition of portraits taken from the cellphones is not always perfect during capture. It’s always difficult to change the lighting condition of the real world when we take photos. Thus, in Chapter 2 we present a system for *single image portrait relighting*: a neural network that takes as input a *single* RGB image of a portrait taken with a standard cellphone camera in an unconstrained environment, and from that image produces a relit image of that subject as though it were illuminated according to any provided environment map. Our method is trained on a small database of 18 individuals captured under different directional light sources in a controlled light stage setup consisting of a densely sampled sphere of lights. Our proposed technique produces quantitatively superior results on our dataset’s validation set compared to prior works, and produces convincing qualitative relighting results on a dataset of hundreds of real-world cellphone portraits. Because our technique can produce a 640×640 image in only 160 milliseconds, it enables interactive user-facing photographic applications. Our proposed algorithm has later led to a new feature *Portrait Light* launched on Google Pixel phones. This feature allows the users to easily alter the lighting condition of a captured portrait photo, as shown in Fig. 1.2.

We use a light stage to capture the appearance of the human subject under different light sources, which enables image-based relighting in novel environments. However, due to the finite

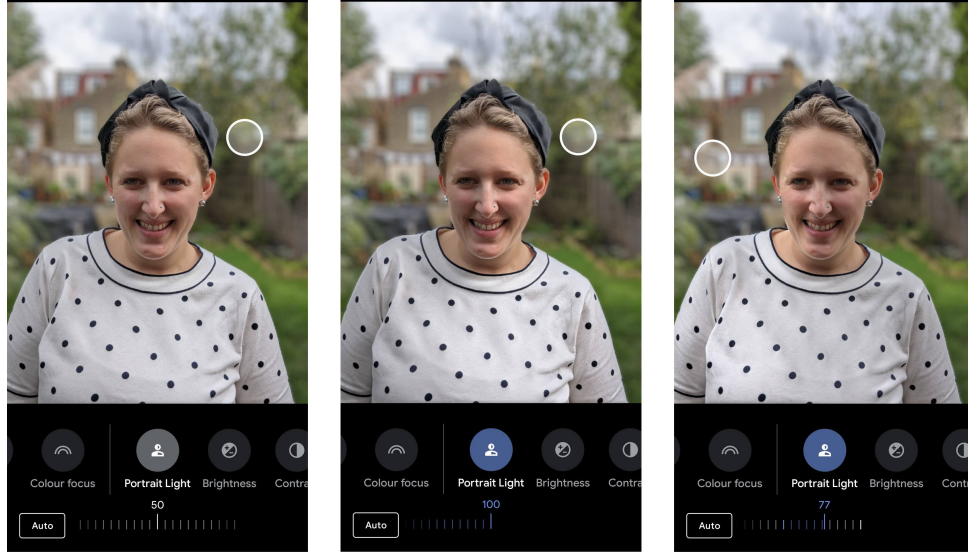


Figure 1.2. Portrait Light on Google Pixel phones

number of lights in the stage, when relighting the subject with a point light or a directional source that does not coincide exactly with one of the lights in the stage, the traditional image-based relighting technique leads to ghosting shadows, aliased specularities, and other artifacts. To ameliorate these artifacts and produce better results under arbitrary high-frequency lighting, in Chapter 3 we introduce a learning-based solution for the “super-resolution” of scans of human faces taken from a light stage. Given an arbitrary “query” light direction, our method aggregates the captured images corresponding to neighboring lights in the stage, and uses a neural network to synthesize a rendering of the face that appears to be illuminated by a “virtual” light source at the query location. This neural network must circumvent the inherent aliasing and regularity of the light stage data that was used for training, which we accomplish through the use of regularized traditional interpolation methods within our network. Our learned model is able to produce renderings for arbitrary light directions that exhibit realistic shadows and specular highlights, and is able to generalize across a wide variety of subjects. Our super-resolution approach enables more accurate renderings of human subjects under detailed environment maps, or the construction of simpler light stages that contain fewer light sources while still yielding comparable quality renderings as light stages with more densely sampled lights.

Human portraits exhibit various appearances when observed not only under different lighting conditions but also from different views. Our relighting algorithms introduced in Chapter 2 and 3 enable changing of the light but not the view. To this end, in Chapter 4 we present a system for portrait view synthesis and relighting: given multiple portraits, we use a neural network to predict the light-transport field in 3D space, and from the predicted Neural Light-transport Field (NeLF) produce a portrait from a new camera view under a new environmental lighting. Our system is trained on a large number of synthetic models, and can generalize to different synthetic and real portraits under various lighting conditions. Our method achieves simultaneous view synthesis and relighting given multi-view portraits as the input, and achieves state-of-the-art results.

In Chapter 5, we summarize the contributions of the thesis and discuss possible future directions on human portrait relighting and view synthesis, and more broadly about appearance representation. We believe our works on portrait relighting and view synthesis have inspired and will continue to inspire new lines of works on general appearance acquisition and representation, including neural rendering approaches, generalized appearance modeling, inverse rendering techniques, dynamic scene acquisition, *etc.*

Chapter 2

Single Image Portrait Relighting

2.1 Introduction

The rise of mobile computing has led to tremendous growth in the popularity of consumer digital photography, and one of the most popular and ubiquitous kinds of photos taken is the portrait: an image of a human subject’s face or upper body. Portrait photography follows in the tradition of portrait painting, where since the renaissance artists have recognized how lighting can capture the depth and essence of the subject on a 2D canvas [99]. These ideas largely influenced professional portrait lighting techniques [98]. In the 18th and 19th centuries, portraits were often taken by professional photographers, who carefully considered and controlled the lighting of the scene in addition to the pose and appearance of their subject. In the modern age of “selfies” and candid photography, it is difficult or impossible for consumer photographers to control the lighting of their subject — one would likely not consider interrupting an engagement proposal or a piano recital to set up studio lighting. Moreover, professional-quality lighting often requires expertise and special equipment such as flashes, deflectors and diffusion panels. These techniques are out of reach for casual mobile phone photographers, whose only available light is usually the flash built into standard consumer cameras, which tends to produce garishly-lit results [86]. These casual photographers would benefit greatly from the ability to *relight* their portrait photos, and change the lighting of their image in a post-processing setting to that of some other environment.



(a) Input image and estimated lighting (b) Rendered images from our method under three novel illuminations

Figure 2.1. Given only a single input image taken with a standard cellphone camera of a portrait (a), our model is able to quickly (160 ms.) generate new images of our human subject as though they are illuminated under new, previously-unseen lighting environments (b).

In this chapter, we present a model that requires as input only a *single RGB image* of a portrait of a single human subject casually captured with a cellphone camera in natural, unconstrained lighting (Figure 2.1a). Our model consists of a single deep neural network that has been trained to take such an image as input and produces as output a relit version of the portrait under an arbitrary user-specified environment map (Figure 2.1b), in addition to predicting the environment map corresponding to the input image (Figure 2.1a).

Unlike traditional [5] and learning-based [101] scene inference or face reconstruction algorithms, our model does not have any explicit inverse rendering step for estimating geometry and reflectance. Such approaches are necessarily limited to what is expressible by their estimated model, such as Lambertian reflectance and spherical harmonic illumination. In contrast, we train a single neural network (Figure 2.2) to directly produce relit images from an input image and a “target” illumination — any representation of geometry and reflectance that our model may recover is fully learned and is represented solely in terms of internal network activations. By training our network to directly predict a final image and by not imposing any physical constraints on the implicit representation of our underlying scene (other than the constraints imposed by neural networks which, given enough capacity, are known to be universal function approximators), our model is capable of capturing most of the subtleties of human facial appearance, such as

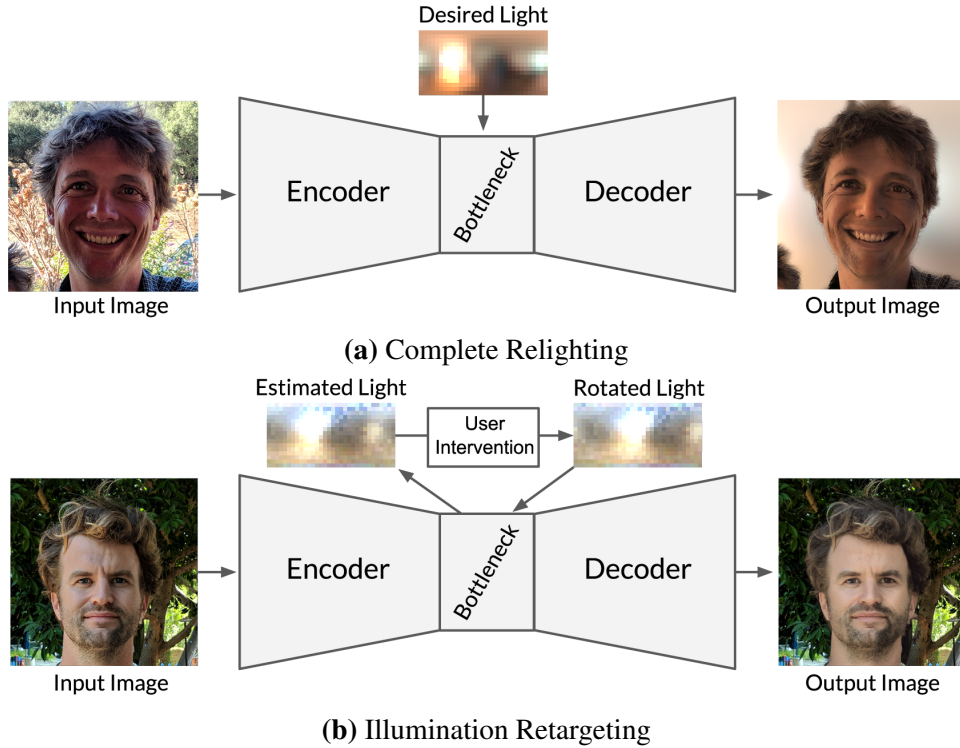


Figure 2.2. Our relighting system is an encoder-decoder neural network that takes as input a single input image and a target illumination (injected into the bottleneck of the network), and produces as output a relit image (a). The encoder also predicts the illumination of the input image, thereby allowing an input image’s illumination to be recovered during encoding, modified within the bottleneck of the network, and then decoded to produce a relit image corresponding to, say, a rotation of the original illumination (b). Images from Google Gcam team members, used with permission.

shading, scattering, specularities, and shadowing.

One could imagine our proposed model as the core component of a user-facing system designed to enable post-capture illumination manipulation on a mobile phone. The user could relight their portraits using various canonical or user-specified environments (Figure 2.2a), similar to the popular “filters” used by social media platforms like Instagram. Or the user could relight their image with the environment map corresponding to the input image produced by our model, by manually rotating or retargeting the environment map to, for example, turn a side-lit scene into a front-lit scene (Figure 2.2b). In this way, our model is superficially similar to the “Portrait Modes” already in use by Apple’s iPhone [2] and Google’s Pixel phone [3, 118] which

use learning and computer vision techniques to enable post-capture manipulation of a camera’s physical depth-of-field. Additionally, the iPhone X features a “Portrait Lighting” mode that adjusts the brightness and contrast of a human subject, and darkens the background. Critically, this feature does not *relight* a scene — it cannot remove or change the illumination of an input image; it can only modify the effects of the existing illumination.

Our method is trained and validated using only real light stage data for just 22 individuals captured from 7 different viewpoints and under 304 lighting directions (18 individuals are used for training, and the remaining 4 are used to validation). By training on real data, we can preserve the subtleties of non-Lambertian reflectance as it relates to the intricate geometry of human faces. This dataset is used to train a novel neural network architecture for portrait relighting, which consists of an encoder-decoder structure that has been augmented to allow for the input image’s illumination to be predicted from the bottleneck of the network, and allows for the target illumination that is to be used for relighting to be injected into the bottleneck of the network. Finally, we demonstrate convincing “in the wild” results on portrait relighting from a single input image, on a dataset of hundreds of casually captured cellphone portraits.

2.2 Related Work

Portrait relighting can be seen as a special case of image relighting, and also relates to intrinsic images and shape from shading, as well as monocular face and scene reconstruction methods, style transfer, photographic post-processing, and deep learning.

Image-Based Relighting

The linearity of light transport allows one to take many images of a subject from the same viewpoint under different known illuminations, and relight the subject simply by weighting and linearly combining those images [19]. Though effective, these techniques require hundreds of images, a near-motionless subject, and precise control over illumination. Therefore, they do not provide a solution when given a single RGB image of an unknown subject in an unconstrained

environment. However, we show that, just as a light stage scan of a person can be used to relight that person under any environment, the data collected from such a light stage can be used for training and evaluating our own portrait relighting algorithm.

Intrinsic Images and Shape from Shading

“Intrinsic image” algorithms, such as the Retinex method [56], abstract away geometry and attribute image content to either shading or reflectance. But these solutions do not enable general purpose relighting, as they do not produce the scene geometry required for rendering novel illuminations. Shape-from-shading solutions assume that materials and illumination are known or fixed and attempt to recover only geometry [40]. But even this constrained problem formulation suffers from a variety of ambiguities [6]. We compare to a recent work [5], which uses priors to estimate shape, illumination, and reflectance from a single image, and we demonstrate that our algorithm has superior relighting performance.

Monocular Scene Understanding

Monocular 3D estimation techniques [38, 95] are capable of recovering scene geometry from a single image, and though such geometry allows for lights to be synthetically *added* to a scene through straightforward rendering techniques, it does not provide a means by which lights can be *removed*, and therefore does not enable *relighting*. Similarly, monocular illumination estimation techniques [55, 30, 39] can recover the environment map that illuminated a single image, but this only enables the lighting of 3D geometry that is added to the scene and does not solve the problem of relighting existing scene content.

Reflectance and Style Transfer

Peers *et al.* [85] showed that a flat-lit facial portrait could be re-lit according to a target environment by multiplying it by an aligned image of a similar reference face seen in the target lighting environment, divided by the appearance of that reference face seen in flat lighting. In this case, high-frequency lighting changes in the skin and hair are transferred from the reference subject, affecting the appearance of hair and specular reflections. The technique

requires a database of reference subjects (chosen manually to approximate the desired subject) to approximate the facial portrait. Shih *et al.* [103] uses a multiscale technique to transfer the local image statistics of a reference facial portrait onto a new one, matching properties such as local contrast and the overall lighting direction. This technique can produce compelling results, but is limited in how much it can change the lighting, and can require manual touch-ups. Nonetheless, both of these techniques establish that believable relighting can be achieved through image processing operations based on reference images of other subjects, an idea which is at the core of our technique as well.

Face Reconstruction and Relighting

Using a morphable model of faces allows for relighting under certain circumstances [10]. But relying on a highly specialized model of faces results in poor performance when presented with non-generic faces or expressions, or with the non-face content that is common in portraits such as glasses, shoulders, or hair. The method of Shu *et al.* [104] builds upon these morphable models to specifically target single-image portrait lighting transfer with compelling results, though the proposed system does not allow for general purpose relighting with any environment map, and shares the aforementioned limitations of a basic morphable model-based approach.

Photographic Post-Processing

Relighting can be thought of as a natural extension of the photometric manipulation already performed by photographic image processing pipelines. Modern digital cameras carefully consider and modify the brightness, exposure, and tone of their output images, all of which can be thought of as the coarse manipulation of scene lighting [36]. Local tone-mapping, which aims to decrease the dynamic range of an input image while preserving its local contrast, can also be thought of as a coarse approximation to relighting, and has been explored through classic algorithms [25, 81] and learning-based approaches [32].

Deep Learning

The revival of end-to-end trained neural networks in the ongoing “deep learning” boom, propelled by compelling results in object recognition [54] and enabled by the availability of differentiable programming frameworks such as Tensorflow [1], has changed the research landscape across computer science. In particular, convolutional neural networks [57] have become critical tools in a variety of works related to relighting. The network proposed by Xu *et al.* [130] is able to learn a relighting function that can reproduce complicated illumination effects, but requires five images captured under predefined directional lights as input. Sengupta *et al.* [101] leverages 3D morphable models as a source of synthetic training data to train a network to regress from a face image to a Lambertian intrinsic decomposition and spherical harmonic illumination, which can then be used for relighting. However, the modeling assumptions imposed by this technique limit the quality of its output renderings, as skin is highly non-Lambertian and real-world environment illuminations are poorly approximated by low-order spherical harmonic illumination. There also exist neural networks [14, 30, 39] that regress from an image (of a face, of an indoor environment, or of an outdoor environment respectively) to the environment that the image was illuminated by, but these approaches do not provide a solution for relighting. Li *et al.* [60] learns to regress from a single image of an object to a shape and spatially-varying BRDF that can then be used to relight that object, but requires a known and constrained illumination in the form of a front-facing flash, thereby preventing user manipulation of any existing natural illumination of the scene, and limiting the system’s utility to situations in which the camera flash is the dominant light source and is not socially-disruptive. In this chapter, we use a neural network to directly learn a function for relighting images in an end-to-end fashion, without explicitly modeling the geometry or the reflectance of the human face.

2.3 Learning Relighting for Portraits

Given a single high-resolution portrait image taken under a natural environment, we want to change only the illumination of the image to another specified lighting environment while keeping the subject of the portrait image the same. We assume that the portrait is taken roughly from the front-facing direction, within ± 20 degrees of deviation from the center, and that the lights are distant from the subject. In addition to a relit image, we want our model to predict the illumination of the input portrait image. Formally, given the “source” portrait image $\mathbf{I}_s$ and the “target” lighting ℓ_t , we want to learn a function $\Phi(\cdot)$ that predicts the lighting of the source portrait image $\hat{\ell}_s$ and a target portrait image $\hat{\mathbf{I}}_t$ lit by the target lighting:

$$\hat{\mathbf{I}}_t, \hat{\ell}_s = \Phi(\mathbf{I}_s, \ell_t). \quad (2.1)$$

We want $\Phi(\cdot)$ to generalize to portraits of a wide range of people with different skin tones, facial expressions, and viewing directions, all taken under a wide variety of different far-field illuminations.

Previous works [5, 101, 44] usually address this problem by first performing an intrinsic decomposition on the image, and then using that decomposition to render a new image according to the target lighting. This approach places implicit constraints on the resulting rendering, often in the form of Lambertian reflectance and spherical harmonic illumination. In this chapter, we adopt an end-to-end approach: we use a neural network to model the relighting function $\Phi(\cdot)$ that directly predicts the portrait under the target lighting. By using learning instead of an explicit representation of a reconstructed scene, our approach is capable of modeling the broader range of physical phenomena present in human skin and faces including interreflection, scattering, shading, and specularities.

In Section 2.3.1, we describe our pipeline for portrait collection and dataset synthesis. The resulting portrait input/output pairs are then used as training data by our Portrait-Relighting

Network, which is described in Section 2.3.2. Our loss function and our training procedure are described in Section 2.3.3.

2.3.1 Data

Portrait Imagery

In contrast to most supervised learning tasks in computer vision, collecting even a small amount of data for a relighting task is onerous: the subject must be stationary, and must be repeatedly imaged from the same viewpoint with different known illuminations. For this reason, previous works [101, 60] have relied heavily on synthetic, often Lambertian, images for use as training data. However, there is a significant gap between real and synthetic faces, particularly in terms of skin reflectance [22] and fine geometric detail [75]. And though morphable models [10] can be used for synthesizing images of faces, they do not provide a means for synthesizing realistic hair, torsos, or accessories. The efficacy of modern deep learning techniques means that, though neural networks are capable of learning a complicated mapping from input to output, they will also absorb the biases and inaccuracies of the data with which they have been trained. Because we want to produce more realistic output than the Lambertian synthetic data commonly used for training, we choose to use only real-world portraits when learning $\Phi(\cdot)$.

We use a light stage setup similar to Wenger *et al.* [125] to capture the real-world portrait images that will be used to synthesize our training data. The light stage consists of 304 LED light sources tessellating a sphere surrounding the subject, and seven machine vision cameras positioned in front of the subject, where all lights and cameras are programmable and synchronized. A human subject is seated in the center of the light stage and is asked to remain stationary. We capture a series of images of the subject in which each of the 304 LED light sources is turned on to a white color while all other lights are off. We refer to this set of images produced by each camera as a “one-light-at-a-time” (OLAT) imageset. The acquisition process lasts about 6 seconds, during which all but the most disciplined of subjects have likely moved slightly. For this reason, during each capture we take an additional image every 11th



(a) OLAT images (7 cameras).

(b) Ground-truth renderings.

Figure 2.3. A set of example images from the hexagon shaped multi-camera rig installed on our light stage for one light in an OLAT imageset (a). The center camera is 1.7 meters away from the subject, and each vertex camera deviates from the center camera by around 20 degrees. This allows us to cover the most common views of a human face in a portrait or selfie. Then, the images can be linearly weighted and combined to obtain images lit by environment illuminations, to generate ground truth for training (b).

frame in which all LEDs are on. These “tracking frames” are used as input to an optical flow algorithm [125], and the resulting flow fields from these frames are used to advect the OLAT frames in the imageset (assuming linear motion between tracking frames) to remove all but the smallest movements made by the subject.

A 7-camera rig (Figure 2.3(a)) in a hexagonal shape is installed on the light stage, and covers a viewing angle for a human face of ± 20 degrees with respect to the frontal view. We capture 22 subjects, each with 3 to 5 different facial expressions (neutral, smile, laugh, frown, and raised eyebrows). We use 18 of the subjects for training and the rest for validation. The

demographics of our subjects were largely determined by circumstance, and as such they are somewhat skewed in terms of skin color and gender. We therefore manually constructed our training and validation set in an effort to prevent overfitting and to avoid under-representing minorities in the validation set. Our training set contains 7 male Caucasians, 7 male Asians, 2 female Caucasians, 1 female Asian, and 1 female of African descent. Our validation set contains 1 male Caucasian, 2 male Asians, and 1 female of African descent. The combination of subjects and facial expressions results in 98 captures, with 81 used for training.

Environment Lighting

To render our OLAT imagesets under natural illumination conditions for training and testing purposes (Figure 2.3(b)), we require many natural illuminations. For this, we use the Laval Indoor HDR Dataset [30] which consists of 2144 high dynamic range indoor environment maps, and we further collect 950 high dynamic range outdoor environment maps taken from a variety of sources: Eisklotz, HDRI Haven, HDRI Skies, HDR Labs, HDRMAPS, NoEmotion HDRs, and Openfootage. In all, we have 3094 different environment illuminations, of which a random selection of 2560 are used for training and the remaining 534 are used for validation.

Data Processing

With our OLAT imagesets and our environment illuminations, we can now generate paired portrait images which differ only in terms of illumination. The resolution of our OLAT images is 2400×1800 . For each image, we first use the human segmentation algorithm [118] to mask out the background. Then we take a random crop from the OLAT imageset, with a position chosen uniformly at random and with the crop size chosen uniformly from $[512, 1024]$ at random. We then randomly select two environment illuminations (represented in latitude-longitude format), and to each we apply a random rotation in $[0^\circ, 360^\circ]$ in its longitude (but not in its latitude). Each environment illumination is then resized to 128×256 , at which point it is projected onto the basis defined by the LEDs of the light stage. By taking a weighted combination of the images in the OLAT imageset according to these projections onto this LED

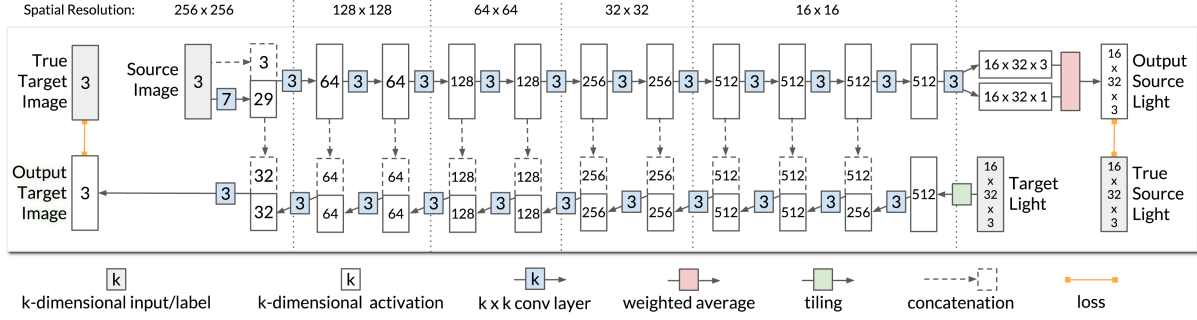


Figure 2.4. The architecture of our neural network. The source input image is passed through a series of conv layers that gradually decrease spatial resolution while increasing the number of channels. After encoding, a confidence-weighted average predicts the illumination that the source image was lit by. The target light is then injected as input into the bottleneck of the network, and this target light along with the encoding of the source image is decoded (with skip connections) into the output target image. Losses are imposed to minimize the differences between the true and output target images, and between the true and output source lights. When evaluating our “self-supervision” loss, this architecture is modified such that the output source light with a certain rotation is used as the target light when decoding.

basis, we are able to render the cropped OLAT imageset under our two lighting conditions. After rendering, these illuminations are then mapped back to the latitude-longitude format of size 16×32 . This is done by modelling each LED as a Gaussian light (where mean position μ is each LED’s position in camera coordinates and standard deviation $\sigma = 8^\circ$) and then projecting onto the basis defined by the PDF of these Gaussians. Finally, the rendered portrait pair is downsampled to 256×256 , and each image (and its corresponding environment illumination) is scaled such that the maximum pixel intensity in each image is 1. All processing is done in the linear domain, without gamma correction. In total, we generate 226,800 portrait pairs as training data. Note that there is no overlap between training or validation subjects or environment illuminations — training illuminations are only ever paired with training OLAT imagesets, and validation illuminations are only ever paired with validation OLAT imagesets.

2.3.2 Network Architecture

The structure of our Portrait-Relighting Network (PR-Net) is shown in Figure 2.2 and Figure 2.4, and superficially resembles the structure of the popular U-Net [93]: an encoder-

decoder structure with skip connections. In the encoder, the source portrait image $\mathbf{I}_s$ is processed by a series of conv layers (with strides of 1 or 2) that gradually decrease spatial resolution and increase the number of channels. The final encoded activations at the bottleneck of the network are then used to predict the illumination corresponding to the source image $\hat{\ell}_s$. This illumination prediction is performed using the “confidence learning” approach [41], which has previously been used for the related problem of color constancy. This approach is used because, due to the finite spatial support of the conv layers in our network, the encoded network activations only reflect a limited region of the whole image — they can only “see” what lies within their receptive field. If the network only observes the part of the image corresponding to the left cheek of subject, it’s unreasonable to ask it to predict the incident illumination coming from the right. However, if we designed our network architecture such that the left side of the illumination were to be predicted solely from the left side of the image, we would be depriving our network of the ability to reason about the entire image, which is valuable given the global nature of illumination. Therefore in our lighting prediction, for each location in our encoding bottleneck (each of which corresponds to some limited receptive field) we have our network predict a complete RGB environment illumination as well as a confidence map associated with that location in the bottleneck. We then take a weighted average of these predictions according to their corresponding confidences to get our final predicted lighting $\hat{\ell}_s$. Note that in order to keep our network fully convolutional, the spatial resolution of the environment illumination is packed into the channel dimension.

In the decoder of our network, we feed the target lighting ℓ_t as input into the network and encode it using conv layers before concatenating it with the representation of the source image produced by the encoder. This concatenated encoding is then passed through a series of transposed conv layers (with strides of 1 or 2), and is gradually upsampled back to a 3-channel image with the spatial resolution of the input source image, which is our output target portrait image $\hat{\mathbf{I}}_t$. At each decoder layer we use a skip connection from the corresponding encoder layer and concatenate the network activations accordingly.

All conv layers are followed by a group normalization [127] and an activation function. A sigmoid activation is used in the layer before the target portrait image prediction (as images are assumed to be in $[0, 1]$), a softplus activation [23] is used immediately before the weighted average that predicts the lighting of the source portrait (as confidence must be non-negative), and a PReLU activation [37] is used after all other layers.

2.3.3 Loss Function and Training

Our model is trained through the minimization of a weighted combination of three loss functions. The first loss minimizes errors between the true target image $\mathbf{I}_t$ in our dataset and the target relit image $\hat{\mathbf{I}}_t$ predicted by our network according to the “target” illumination ℓ_t . The second loss minimizes errors between the predicted source illumination $\hat{\ell}_s$ and the true source illumination ℓ_s . To evaluate our third “self-supervision” loss, we modify our network architecture slightly and replace the true target illumination ℓ_t that was previously used as input to the decoder stream of the network with the predicted source illumination $\hat{\ell}_s$ that is produced by the model, and minimize errors between the resulting reconstruction of the source image $\hat{\mathbf{I}}_s$ and the input source image $\mathbf{I}_s$. The goal of the first loss is straightforward: we would like our model to produce accurate relightings. The next two losses have a more nuanced effect: by ensuring that the model can accurately reproduce its own input image and illumination, we enable the use-case in which our model is used to alter the existing illumination in the scene, rather than replace it entirely with some new illumination. Additionally, we will show that imposing these two secondary loss functions actually improves performance on our primary goal of relighting.

The loss we will use when comparing relit or reconstructed images is the per-pixel L_1 distance between our predicted image $\hat{\mathbf{I}}$ and the true image $\mathbf{I}$, after the background has been masked out:

$$\mathcal{L}_1(\hat{\mathbf{I}}, \mathbf{I}) = \|\mathbf{M} \odot (\hat{\mathbf{I}} - \mathbf{I})\|_1, \quad (2.2)$$

where $\mathbf{M}$ is a per-pixel portrait mask, and $\odot$ is element-wise multiplication. For the illumination

prediction, we use the weighted log- L_2 distance of [123]:

$$\mathcal{L}_L(\hat{\ell}, \ell) = \|\Omega \odot (\log(1 + \hat{\ell}) - \log(1 + \ell))\|_2^2, \quad (2.3)$$

where Ω is the solid angle of each “pixel” in our environment illumination in their latitude-longitude representation. With these defined, we can construct our complete loss function as:

$$\mathcal{L} = \mathcal{L}_1(\hat{\mathbf{I}}_t, \mathbf{I}_t) + \lambda_{light} \cdot \mathcal{L}_L(\hat{\ell}_s, \ell_s) + \lambda_{self} \cdot \mathcal{L}_1(\hat{\mathbf{I}}_s, \mathbf{I}_s), \quad (2.4)$$

where we set the hyperparameters $\lambda_{light} = 0.8$ and $\lambda_{self} = 1$.

As is often the case with autoencoders, our self-supervision loss works best if the predicted source illumination that is fed back into the decoder of the network is “jittered” slightly. Otherwise the network may begin to generalize poorly to illuminations that are similar to training data instances but are not themselves present in the data, which results in flickering when synthetically rotating the illumination during relighting and in color shifts when relighting in general. To ameliorate this, during training we apply a random rotation of $\theta \in [0^\circ, 360^\circ]$ to each environment along its longitude, and correspondingly re-render the “source” image with that rotated environment.

Our model is implemented using TensorFlow [1]. We train our model on 4 NVIDIA Tesla V100s with batch size 2 on each device. We use the Adam optimizer of [53] with a learning rate of 10^{-5} to train the network, and our model converges after 20 epochs (which takes ~ 26 hours).

2.4 Evaluation

Here we evaluate our proposed model against prior works and against ablations of our own model. We present results on two tasks: in Section 2.4.1 we measure the quality of the relit images produced by each model, and in Section 2.4.2 we measure the quality of the environmental

illuminations produced by each model.

2.4.1 Relit Images

Metrics

We measure the relighting performance using three error metrics across our validation set: RMSE, scale-invariant RMSE (RMSE-s), and DSSIM. Our scale-invariant RMSE solves for the single scale-factor that can be applied to the predicted image to best minimize RMSE.

$$\text{RMSE-s}(\hat{\mathbf{I}}_t, \mathbf{I}_t) = \min_{\alpha} \|\alpha \hat{\mathbf{I}}_t - \mathbf{I}_t\|_2 \quad (2.5)$$

We use this metric because a relit image that is correct up to a scale factor may still be useful in many photographic contexts. Note that this metric solves for a single global scaling rather than a per-channel scaling, meaning that it is still sensitive to erroneous tints in the output image. Our DSSIM implementation uses a 11×11 Gaussian filter with $\sigma = 1.5, k_1 = 0.01, k_2 = 0.03$, as is recommended by [121]. DSSIM is computed on each RGB channel individually, which (because DSSIM is invariant to local scaling) means that our DSSIM metric is invariant to both global and local scaling and color shifts. Our metrics thereby span a continuum of invariance, from RMSE (which is invariant to nothing) to DSSIM (which is invariant to local and global scaling and tinting). This continuum is worth considering when designing error metrics for this task, as a common mistake for relighting or intrinsic image decomposition algorithms is that of erroneously producing a scaled up illumination and a correspondingly scaled down albedo.

Intrinsic decomposition methods

In Table 2.1 we present the performance of our models compared to several state-of-the-art methods for the single-image relighting task on the validation set of our dataset. This table primarily serves to report the performance of each algorithm in terms of prediction of the “target” portrait image for each validation set scene, though we additionally report the accuracy of each model in reconstructing the source image — the input image itself — according to the predicted

Table 2.1. Here we benchmark our model on the validation set of our dataset against prior work on single-image relighting and against ablations of our model. The “Target” metrics measure each algorithm’s accuracy in predicting each desired relit image, and the “Source” metrics measure each algorithm’s accuracy in predicting the input image itself according to the model’s predicted illumination. The reported metrics and runtimes are the arithmetic mean over all images in the validation set. The top three techniques for each metric are highlighted in red, orange, and yellow respectively.

Algorithm	Target			Source			Time (sec)
	RMSE	RMSE-s	DSSIM	RMSE	RMSE-s	DSSIM	
SIRFS [5]	0.1715	0.0935	0.0827	0	0	0	788
SfSNet [101]	0.2397	0.0997	0.1297	0.0698	0.0673	0.0899	0.7442
Our model	0.0435	0.0351	0.0251	0.0170	0.0156	0.0060	0.1616
Ours w/o light prediction	0.0460	0.0395	0.0251	-	-	-	-
Ours w/o self supervision	0.0448	0.0370	0.0258	0.0313	0.0282	0.0118	-

illumination from the model. These “source” error metrics are of interest for the use-case in which a user of our system wishes to modify the illumination of the input image, rather than completely replace it. Note that it is possible for a model to trivially minimize the “source” error metrics by simply returning the input image, but it is not possible to trivially minimize the “target” error metrics. In addition to these error metrics, we additionally report the average runtimes for each technique across the validation set, all of which were benchmarked using the same computer that was used during training.

Our comparison against SIRFS [5] was done using the code provided by the authors, with no modification. Relighting is performed by taking the surface normals and reflectance recovered by the model along with the desired environment illumination, then solving a linear system to recover the spherical harmonic illumination that best reproduces the log of the environment illumination (this model is designed to work with “log-shading”), and then rendering the surface normals and reflectance according to that spherical harmonic illumination. Note that the source error metrics are trivially minimized by this baseline, because the intrinsic decomposition produced by this model is constrained to exactly reproduce the input image by construction.

Because the SIRFS supports arbitrary-resolution inputs, images are processed at their native resolution, and the output relit images are downsampled to the 640×640 resolution used by our model. Our comparison against the SfSNet [101] was also done using the code provided by the authors with no modification, and relighting was performed by rendering the output normals and reflectance produced by the model with its predicted environment illumination. Because this model only supports 128×128 resolution inputs, we downsample the input image before relighting and then upsample the output relit image to 640×640 before computing our error measures.

From Table 2.1 we see that our model outperforms prior work in terms of all three of our “Target” error metrics by a significant margin, with reduced error rates compared to the next best-performing technique by 57% - 75%. Our performance on the “Source” error metrics is the second lowest after the SIRFS [5], which by construction perfectly reproduces its input image. Additionally, our technique is $\sim 4.6\times$ faster than the SfSNet [101] and $\sim 4800\times$ faster than the SIRFS [5]. To further demonstrate this improved performance, in Figure 2.5 we visualize our model’s performance compared to our baselines on three of our validation set scenes, and see that our model consistently produces realistic relighting results while the baseline techniques do not. Figure 2.5 also includes a comparison against the technique of Li *et al.* [60], which (unlike our technique and our primary baselines) requires an input image in which the illumination is fronto-parallel. To produce these baseline results we use the fronto-parallel illuminated image from each OLAT imageset as input to this model, thereby satisfying its constraints. This is a generous comparison, as a known and controlled illumination significantly reduces the difficulty of single-image relighting. Despite this advantage, as shown in Figure 2.5, this approach does not produce satisfactory results on this task.

Lighting transfer methods

In Table 2.2 we compare our model against two illumination *transfer* algorithms [103, 104], which do not perform relighting in the same manner as our model, but instead work by

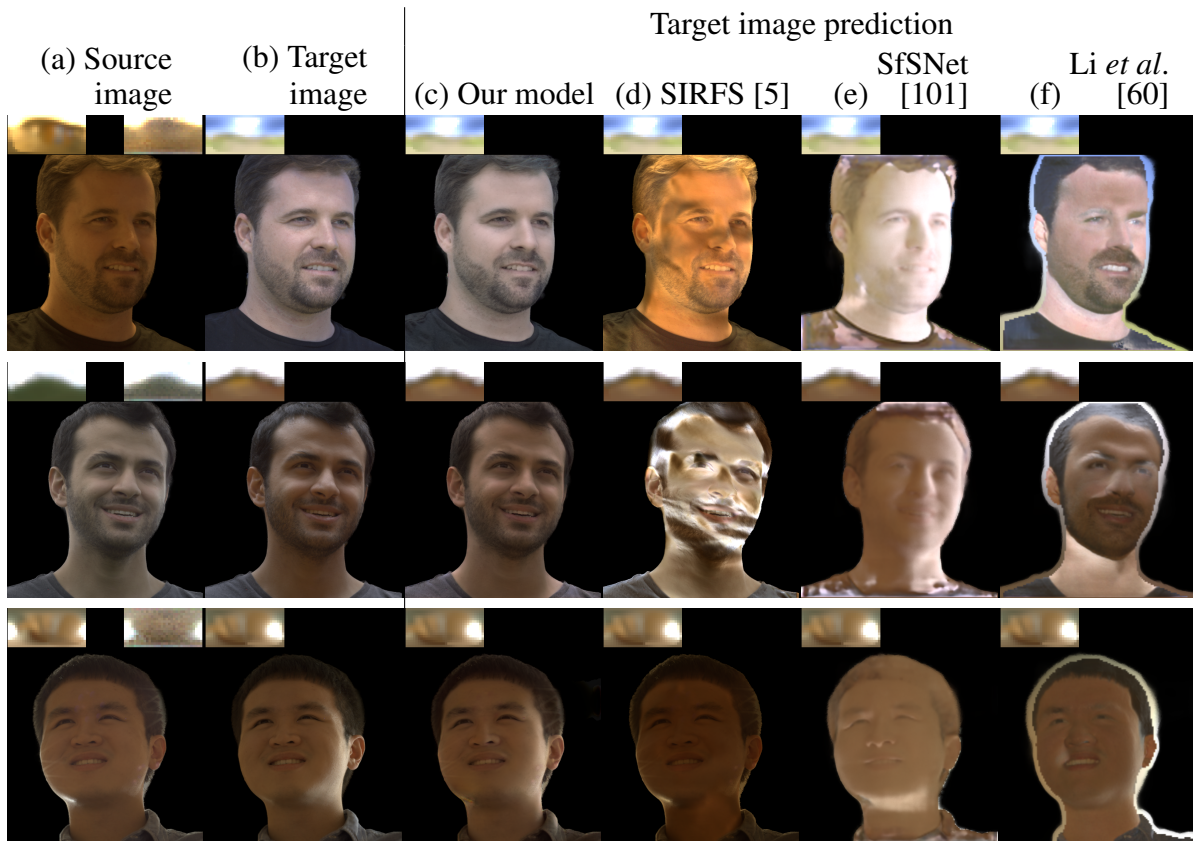


Figure 2.5. Here we present results using the validation set of our dataset. The “Source image” (a) shows the input image to our system and the true illumination (top left) that lit this scene, as well as the illumination predicted by our model (top right). The “Target image” (b) shows the true illumination (top left) that we wish to relight the “source” image with, as well as the true appearance of the subject under that illumination. The remaining columns compare the output of our model (c) with three state-of-the-art baseline models (d, e, f). Our model’s output is significantly more realistic and compelling than that of any baseline, even when presented with challenging scenarios such as relighting from a back-lit scene to a front-lit scene (row 1), cast shadows (row 2), and subsurface-scattering and translucency due to backlighting (row 3).

transferring the lighting conditions from one portrait to another. These transfer algorithms take as input two images: the “source” image that is to be modified, and a “reference” image that contains the illumination that we would like to transfer. The output of these models should be the subject of the source image lit by the illumination in the reference image (the “target” image, using the nomenclature of this work).

Comparing our model against these transfer techniques is challenging, as they do not learn from training data, and instead require a reference image as input. Therefore, when evaluating

Table 2.2. Here we compare our model against two lighting transfer algorithms. Each transfer algorithm is evaluated in two settings: a generous setting one in which the ground-truth output image is presented as input as a reference, and a more-fair setting in which the reference image is a portrait of a different subject illuminated by the desired lighting. Our model significantly outperforms the two baselines in the randomized setting, and even outperforms the two baselines on two of our three metrics in the “oracle” setting where the baselines have access to the ground-truth

Algorithm	Target		
	RMSE	RMSE-s	DSSIM
Shih <i>et al.</i> [103] + ground-truth	0.0406	0.0401	0.0373
Shu <i>et al.</i> [104] + ground-truth	0.0563	0.0528	0.0408
Shih <i>et al.</i> [103]	0.1012	0.0874	0.0653
Shu <i>et al.</i> [104]	0.0697	0.0456	0.0300
Our model	0.0435	0.0351	0.0251

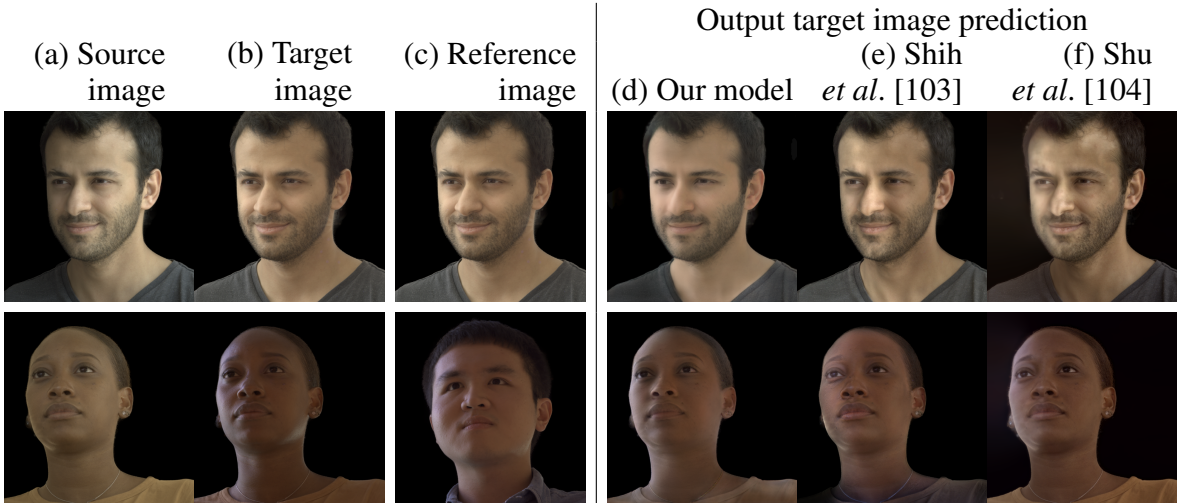


Figure 2.6. Here we show a qualitative comparison between lighting transfer algorithms our method. In the top row we provide the target image as a reference to each lighting transfer algorithm, and in the bottom row we provide a random subject as a reference (these reference images are not used by our model). Under both setups, our model produces a more natural lighting on the portrait, while both baseline lighting transfer methods introduce artifacts on human faces.

against these models on our validation set, we must decide which image from our training data to present to each model as its “reference”. In an attempt to produce a fair comparison, we primarily evaluate against each baseline by using a random image of a different person rendered under

the “target” illumination as the reference image. In this setting, we outperform both baselines on all error metrics by a large margin. We also present additional “oracle” results in which we use the ground-truth target image directly as the reference image. This comparison is generous to the transfer techniques, as simply returning the reference image would result in a perfect reconstruction of the target image with zero error. Performance in this “oracle” setting should be thought of as a bound on performance in more realistic settings. Despite the advantage of the transfer techniques in this setting, our model still outperforms these two “oracle” baseline techniques in two of the three error metrics we evaluate against. See Figure 2.6 for a qualitative comparison of our model and these lighting transfer algorithms.

Ablation study

We additionally perform an ablation study of our own model to demonstrate the contribution of specific model components. In the model labeled “Ours w/o Light Prediction” in Table 2.1 we set $\lambda_{light} = 0$, thereby disabling the loss on the branch of our network architecture that would otherwise predict the illumination of the input image. As such, this model is unable to reconstruct its own input image (hence the missing “Source” error metrics for this baseline), but we also see that this model performs worse on the “Target” metrics than our complete model, thereby demonstrating this additional illumination supervision improves performance on the relighting task. In the model labeled “Ours w/o Self-Supervision” we set $\lambda_{self} = 0$, thereby disabling the self-supervision loss that would otherwise encourage training to accurately reconstruct the source input image. This decreases the accuracy of the model in terms of the source error metrics, as one would expect, but it also reduces performance in terms of the target error metrics, thereby demonstrating that self-supervision is a useful cue even if accurately reproducing the input image is not a desired goal of the model. See Figure 2.7 for a visualization of the effect of self-supervision.

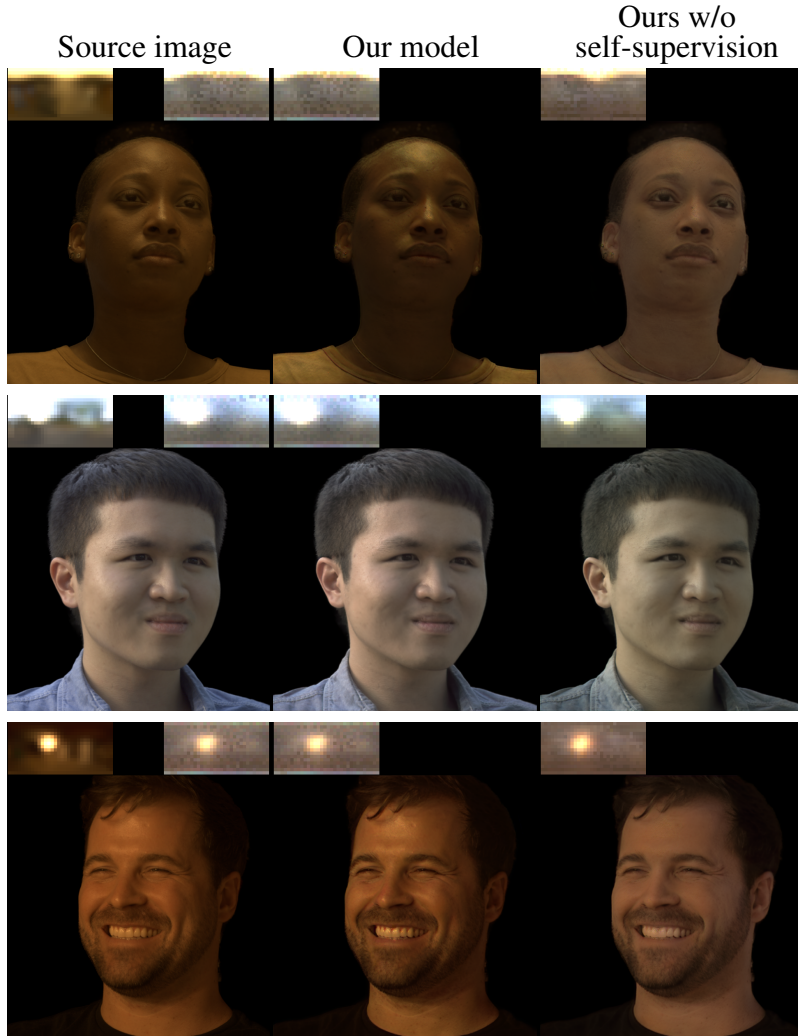


Figure 2.7. Instead of completely replacing the true illumination of the scene (left top inset), one may want to instead recover the existing illumination of the scene (right top inset) and then modify it. To this end, we impose a self-supervision loss during training, which causes our model (middle column) to more accurately reproduce the input (left column) than a model trained without self-supervision (right column). Self-supervision helps our model correctly disentangle skin color from illumination color, as evidenced by the unnatural skin colors in the model without self-supervision.

2.4.2 Estimated Lighting

We also evaluate our model’s ability to predict the lighting from a single portrait. Due to the ambiguity between light source strength and surface albedo [6], we use scale-invariant

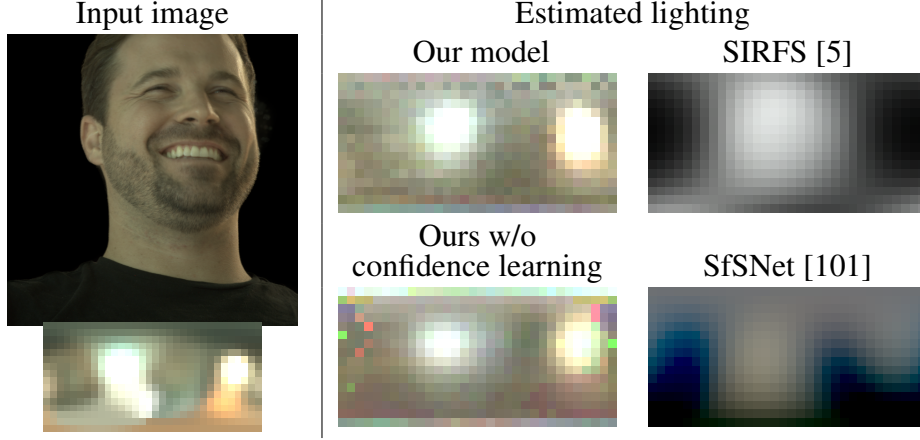


Figure 2.8. Here we show lighting estimation results for our model and prior works given a single portrait image. Our model can accurately recover the locations and the colors of multiple light sources, while other methods struggle with high-frequency illumination effects or the non-Lambertian properties of human skin.

Table 2.3. Here we evaluate our model in terms of the quality of its estimated illumination environments, compared to previous works and to our model without confidence learning. Our model outperforms prior lighting estimation algorithms, and we see that removing confidence learning significantly degrades performance, thereby demonstrating its value.

Algorithm	RMSE-s
SIRFS [5]	1.3972
SfSNet [101]	1.3252
Our model	0.6633
Our model w/o confidence learning	0.8231

RMSE as our evaluation metric, weighted by the solid angle:

$$\text{RMSE-s}(\hat{\ell}, \ell) = \min_{\alpha} \left\| \Omega \odot (\alpha \hat{\mathbf{I}}_{\mathbf{t}} - \mathbf{I}_{\mathbf{t}}) \right\|_2. \quad (2.6)$$

Where Ω contains the solid angle of each pixel in the environment illumination.

In Table 2.3 we evaluate the predicted illuminations produced by our model against two other lighting estimation algorithms [5, 101], as well as an ablation of our model without confidence learning. Our method outperforms our ablation and all prior works by a large amount. See Figure 2.8 for a visualization of these predicted illuminations. The poor performance of the

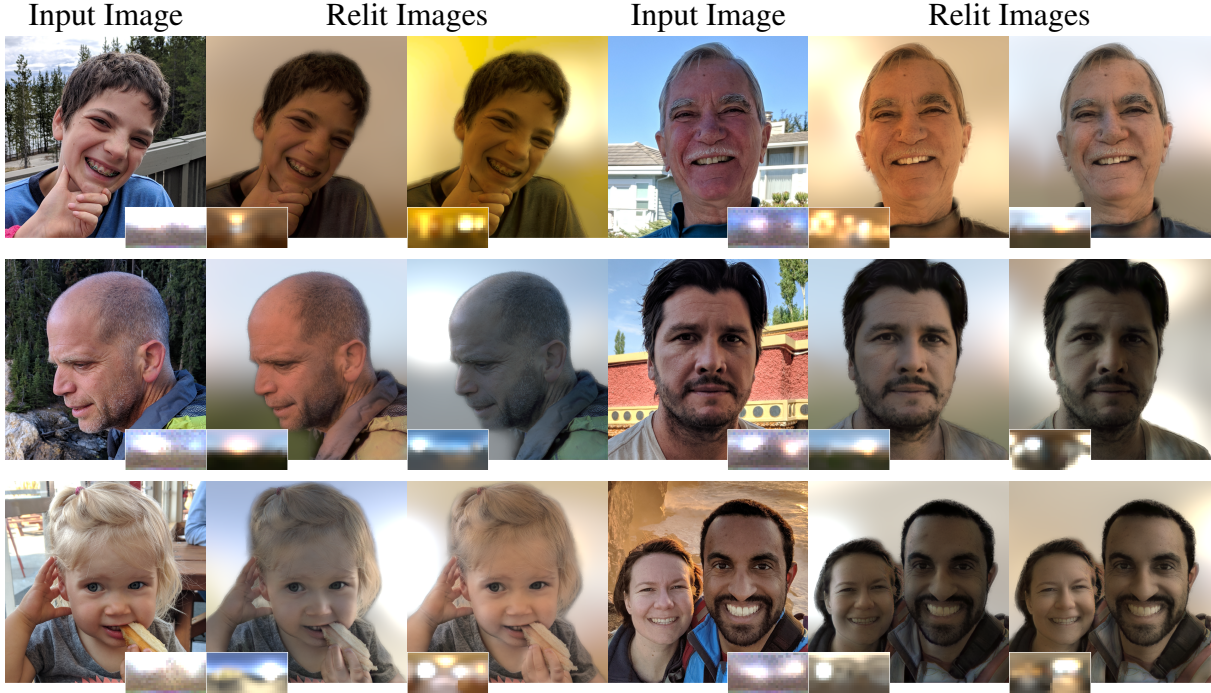


Figure 2.9. Examples of our model being used for complete relighting on “in the wild” portrait images taken from the Portrait-mode and HDR+ dataset [118, 36]. Using only an RGB input image and the portrait mask produced by [118], we are capable of producing relit images according to arbitrary environment illuminations for a wide variety of subjects and viewing angles. First and third input images in the left column are from Google Gcam team members, used with permission.

intrinsic decomposition methods [5, 101] is likely due to their use of spherical harmonics to model illumination, which constrains their output to model only low frequency effects. Confidence learning’s positive contributes is likely because it allows the network to reason about only a part of the lighting from an image patch, thus it helps reducing the speckling in the predicted light behind human subject.

2.5 Real-world Results

We additionally evaluate our model on a wide variety of portraits image taken with a standard cell phone, using the Portrait-mode and HDR+ dataset [118, 36]. Despite the small number of subjects in our training data, our model generalizes well to these varied images taken

“in the wild”.

Complete relighting

Figures 2.1 and 2.9 show the results of using our model for the task of complete relighting — replacing the illumination of the scene. The foreground masks produced by Wadhwa *et al.* [118] are used to isolate the portrait’s subject before it is used as input to our model, and our renderings are produced by compositing the relit foreground over a synthetic rendering of the target illumination. Despite significant variation in the skin tones and ages of the subjects, and in the camera angle and initial illumination, our model is capable of producing realistically relit renderings. Our model is able to process input images that exhibit complex appearance effects such as specularities, soft-shadows and sub-surface scattering, and successfully reproduces those effects in its output renderings. While our network is trained with only viewing angles within a 20° cone around the frontal direction, our method appears to generalize to images with more slanted viewing directions (e.g. the second row of Figure 2.9). Our method also generalizes in terms of the age of the subject, as though our training data consists only of people aged 20 through 40, we are able to process images of children and seniors (e.g. the last two rows of Figure 2.9).

In Figures 2.1 and 2.10, we see that our network is capable of handling portraits in which the subject contains objects that are not present in the training set (food, hats, and sunglasses). The network appears to relight and recolor these objects in a way similar to skin or clothing, which results in plausible looking renderings.

Illumination retargeting

Because our model produces an estimation of the illumination of the input image, and because of our use of self-supervision during training, our model can be used to “retarget” the input illumination and render new portraits in which the original lighting is rotated, as shown in Figure 2.11. When rendering these examples, instead of replacing the background of the image with the environment map, we composite our relit image back onto the background of the input



Figure 2.10. Portrait images often contain subjects holding objects or wearing accessories such as sunglasses. Despite not having seen such objects and accessories during training, our model generalizes reasonably to these images.

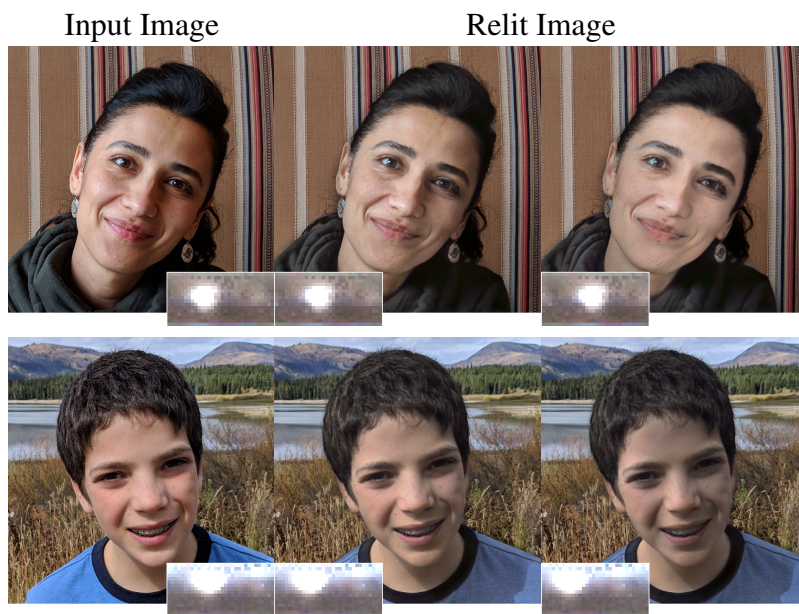


Figure 2.11. Here we present instances of illumination retargeting on “in the wild” portrait images. The illumination of the input image is recovered by our model, manually rotated by the user, and then this retargeted illumination is used to re-render the image. Second input image is from Google Gcam team members, used with permission.

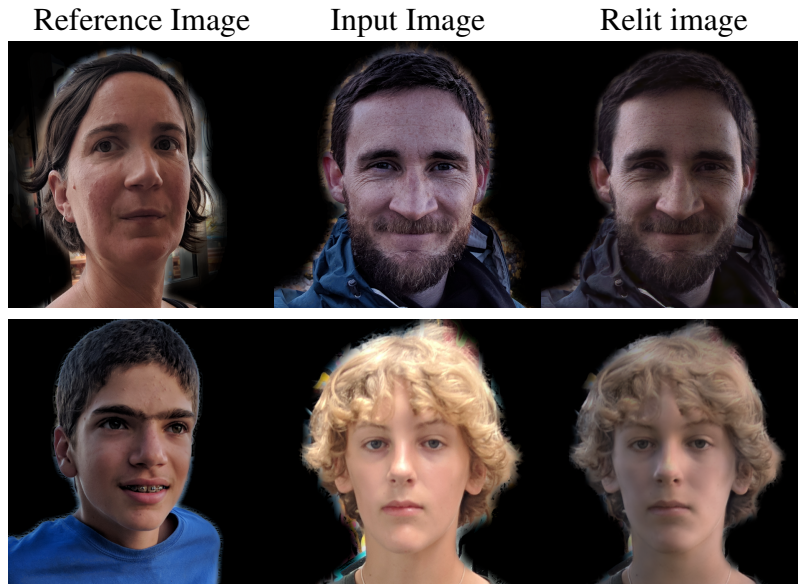


Figure 2.12. Our model can be used for lighting transfer, by using the predicted light of a reference image to relight some input image. First reference image is from Google Gcam team members, used with permission.

image using the same mask used to extract the foreground subject.

Light transfer

Though our model was not designed for the task of transferring the lighting from one portrait to another, it can be used for lighting transfer: we predict the lighting from one portrait and then relight another image according to that predicted lighting. As we can see in Figure 2.12, despite different hair colors or facial expression, our algorithm is capable of producing reasonable lighting transfer results.

Limitations

While our method performs well in most cases, it is not without limitations, as can be seen in Figure 2.13. Our model generalizes poorly when presented with input images that contain hard shadowing, sharp specularities, or saturated pixels, as these phenomena are underrepresented in our training data. It might be possible to fix this problem by adding more training images with hard shadows and sharp specularities. Additionally, our model struggles when presented with accessories containing saturated colors, which are not present in our training data. It may be

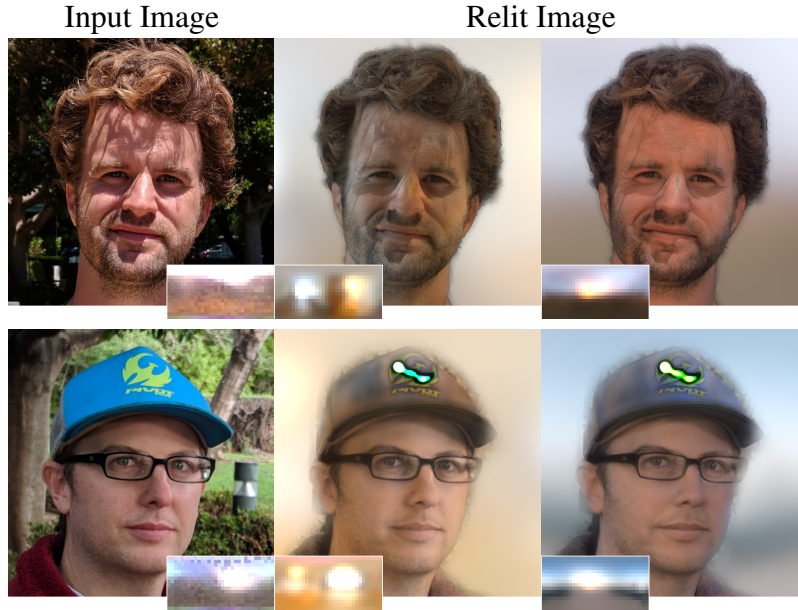


Figure 2.13. When presented with images which deviate significantly from our training data, our model may begin to generalize poorly. Input images lit under harsh illuminations that result in hard shadows, specularities, or saturated pixels may result in artifacts in the output renderings (top). Additionally, subjects wearing accessories with highly saturated colors may result in artifacts (bottom). Images are from Google Gcam team members, used with permission.

possible to use semantic or geometric understanding to ameliorate this issue.

2.6 Conclusion

In this chapter, we have presented a learning-based technique for single image portrait relighting: taking a single RGB image of a human subject in an unconstrained environment and modifying it to appear as though it were illuminated by a different environment. To train our model we have captured a light stage database of a small number of individuals imaged under directional light sources which, when combined with abundantly-available natural environment maps, allows us to generate a large amount of realistic training data for relighting. We have presented a novel neural network architecture for predicting the illumination of the original environment alongside a relit output image in a new target environment, and we have demonstrated the value of self-supervision during training. Our technique outperforms state-of-

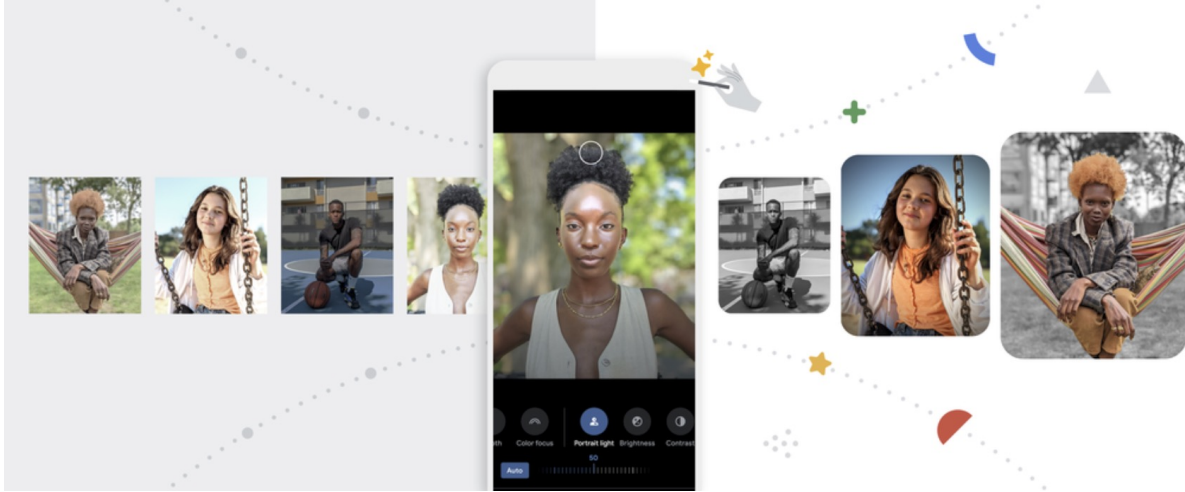


Figure 2.14. Based on our technology of single image portrait relighting, Google has introduced Portrait Light into Pixel phones as well as other applications such as Google Photo.

the-art relighting techniques on the validation set of our dataset, and performs well in practice on another dataset of hundreds of real-world unconstrained portrait photos taken with an ordinary cellphone camera.

The high-quality renderings produced by our technique, combined with its generality and speed (160 milliseconds per 640×640 image) have enabled compelling consumer-facing photographic relighting applications. Based on the technology introduced in this Chapter, Google has launched a new feature *Portrait Light* on Pixel phones, which allows the consumers to alter the lighting condition of a portrait after the capture by just tapping their fingers on the screen. This technology has also been extended to other Google products such as Google Photo (Fig. 2.14). We envision this technology of single image portrait relighting will be more prevalent in future commercial products, enabling everyone to easily control the lighting of portraits.

This chapter is a reformatted version of the material as it appears in “Single Image Portrait Relighting”, Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E. Debevec, and Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 38 (4), 2019 [109]. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Light Stage Super-Resolution: Continuous High-Frequency Relighting

3.1 Introduction

The portrait relighting algorithm presented in Chapter 2 enables changing the lighting of a portrait into natural, low-frequency illuminations. However, the algorithm is not capable of handling all kinds of difficult lightings, such as point lights or a very bright sun. In this Chapter, we investigate how to control the lighting condition of a portrait under all possible illuminations, especially under high-frequency lightings. This is actually a central problem in computer graphics and computer vision: acquiring some observations of an object, and then producing photorealistic relit renderings of that object.

A light stage represents an effective solution for this task: by programmatically activating and deactivating several LED lights arranged in a sphere while capturing synchronized images, the light stage acquires a full reflectance field for a human subject, which we refer to as a “one-light-at-a-time” (OLAT) image set. Because light is additive, this OLAT scan represents a lighting “basis”, and the subject can be relit according to some desired environment map by simply projecting that environment map onto the light stage basis [19].

Though straightforward and theoretically elegant, this classic relighting approach has a critical limitation. The lights on the light stage are usually designed to be small and distant from the subject, so that they are well-approximated as directional light sources. As a consequence,



(a) $\mathbf{I}_1$ and $\mathbf{I}_2$, captured light stage images for adjacent lights ℓ_1 and ℓ_2 (b) Blended images $(\mathbf{I}_1 + \mathbf{I}_2)/2$ (c) Our rendering $\mathbf{I}((\ell_1 + \ell_2)/2)$

Figure 3.1. Though the light stage is a powerful tool for relighting human subjects, its renderings suffer because adjacent lights of the stage are separated by some distance (a). Using conventional image blending techniques to reconstruct the image corresponding to a “virtual” light that lies between the stage’s actual lights therefore results in ghosting in shadowed and specular regions (b), seen here on the subject’s eyes and cheek. By training a deep neural network to regress from a light direction to an image, our model is able to synthesize accurate renderings of the subject under arbitrary virtual light directions — as the light moves, highlights and shadows move smoothly instead of incorrectly blending together, thereby enabling realistic high-frequency relighting effects (c). These images have been manually but uniformly brightened and color-corrected, and are rendered with insets to highlight detail.

realistic high-frequency effects such as sharp cast shadows and specular highlights are present in the captured OLAT images. In order to achieve photorealistic relighting results under *all* possible lighting conditions, the lights must be placed closely enough on the sphere of the stage such that shadows and specularities in the captured images of adjacent lights “move” by less than one pixel. However, practical constraints (the cost and size of each light, and the difficulty of powering and synchronizing many lights) discourage the construction of light stages with very high densities of lights. Even if such a high-density light stage could be built, the time to acquire an OLAT increases linearly with the number of lights, and this makes human subjects (which must be stationary during OLAT acquisition) difficult to capture. For these reasons, even the most sophisticated light stages in existence today contain only a few hundred lights that are spaced many degrees apart. This means that the OLAT scans from a light stage are *undersampled* with respect to the angular sampling of lights, and the rendered images using conventional approaches will likely contain *ghosting*. Attempting to render an image using a “virtual” light source that

lies in between the real lights of the stage by applying a weighted average on adjacent OLAT images will not produce a soft shadow or a streaking specularity, but will instead produce the superposition of multiple sharp shadows and specular dots (see Fig. 3.1b).

This problem can be mitigated by imaging subjects that only exhibit low-frequency reflectance variation, or by performing relighting using only low-frequency environment maps. However, most human subjects have complicated material properties (specularities, scattering, *etc.*) and real-world environment maps frequently exhibit high-frequency variation (bright light sources at arbitrary locations), which often results in noticeable artifacts as shown in Fig. 3.1b. To this end, we propose a learning based solution for super-resolving the angular resolution of light stage scans of human faces. Given an OLAT scan of a human face with finite images and the direction of a desired “virtual” light, our model predicts a complete high-resolution RGB image that appears to have been lit by a light source from that direction, even though that light is not present in our light stage (see Fig. 3.1c). Our robust solution for “upsampling” the number of lights, which we refer to as *light stage super-resolution*, can additionally enable the construction of simpler light stages with fewer lights, thereby reducing cost and increasing the frame rate at which subjects can be scanned. Our algorithm can also produce better rendered images for applications that require light stage data for training, such as portrait relighting or shadow removal. Casual users can then utilize these algorithms on a single cellphone without requiring capture inside a light stage. Note that we focus only on human face relighting within a light stage. While we believe the methods herein could be applied more broadly, a comprehensive system for general object relighting remains a topic of future work.

Our algorithm (Sec. 3.3) must work with the inherent aliasing and regularity of the light stage data used for training. We address this by combining the power of deep neural networks with the efficiency and generality of conventional linear interpolation methods. Specifically, we use an active set of closest lights within our network (Sec. 3.3.1) and develop a novel alias-free pooling approach to combine their network activations (Sec. 3.3.2) using a weighting operator guaranteed to be smooth when lights enter or exit the active set. Our network allows us to

super-resolve an OLAT scan of a human face: we can take our learned model and repeatedly query it with thousands of light directions, and treat the resulting set of synthesized images as though they were acquired by a physically-unconstrained light stage with an unbounded sampling density. As we will demonstrate, these super-resolved “virtual” OLAT scans allow us to produce photorealistic renderings of human faces with arbitrarily high-frequency illumination content.

3.2 Related Work

The angular undersampling from the light stage relates to much work over the past two decades on a frequency analysis of light transport [89, 94, 24], and can also be related to analyses of sampling rate in image-based rendering [15] for the related problem of view synthesis [72]. This problem also bears some similarities to multi-image super-resolution [71] and angular super-resolution in the light field [43, 17], where aliased observations are combined to produce interpolated results. In this Chapter, we leverage priors and deep learning to go beyond these sampling limits, upsampling or super-resolving a sparse input light sampling on the light stage to achieve continuous high-frequency relighting.

Recently, many approaches for acquiring a sparse light transport matrix have been developed, including methods based on compressive sensing [84, 100], kernel Nystrom [119], optical computing [80] and neural networks [92, 90, 45]. However, these methods are not designed for the light stage and are largely orthogonal to our approach. They seek to acquire the transport matrix for a fixed light sampling resolution with a sparse set of patterns, while we seek to take this initial sampling resolution and upsample or super-resolve it to much higher-resolution lighting (and indeed enable continuous high-frequency relighting). Most recently, [130] proposed a deep learning approach for image-based relighting from only five lighting directions, but cannot reproduce very accurate shadows. While we do use many more lights, we achieve significantly higher-quality results with accurate shadows.

The general approach of using light stages for image-based relighting stands in contrast

to more model-based approaches. Traditionally, instead of super-resolving a light stage scan, one could use that scan as input to a photometric stereo algorithm [126], and attempt to recover the normal and the albedo maps of the subject. More advanced techniques were developed to produce a parametric model of the geometry and reflectance for even highly specular objects [115]. There are also works that focus on recovering a parametric model from a single image [101], constructing a volumetric model for view synthesis [62], or even a neural representation of a scene [112]. However, the complicated reflectance and geometry of human subjects is difficult to even parameterize analytically, let alone recover. Though recent progress may enable the accurate capture of human faces using parametric models, there are additional difficulties in capturing a complete portrait due to the complexity of human hair, eyes, ears, etc. Indeed, this complexity has motivated the use of image-based relighting via light stages in the visual effects industry for many years [116, 18].

Interpolating a reflectance function has also been investigated in the literature. Masselus *et al.* [66] compare the errors of fitting the sampled reflectance function to various basis functions and conclude that multilevel B-Splines can preserve the most features. More recently, Rainer *et al.* [88] utilize neural networks to compress and interpolate sparsely sampled observations. However, these algorithms interpolate the reflectance function independently on each pixel and do not consider local information in neighboring pixels. Thus, their results are smooth and consistent in the light domain, but might not be consistent in the image domain. Fuchs *et al.* [27] treat the problem as a light super-resolution problem, and is the most similar to our work. They use heuristics to decompose the captured images into diffuse and specular layers, and apply optical-flow and level-set algorithms to interpolate highlights and light visibility respectively. This approach works well on highly reflective objects, but as we will demonstrate, it usually fails on human skin which contains high frequency bumps and cannot be well modeled using only diffuse and specular terms.

In recent years, light stages have also been demonstrated to be invaluable tools for generating training data for use in deep learning tasks [69, 35, 109, 77]. This enables user-

facing effects that do not require acquiring a complete light stage scan of the subject, such as “portrait relighting” from a single image introduced in Chapter 2 or VR experiences [35]. These learning-based applications suffer from the same undersampling issue as do conventional uses of light stage data. For example, we can observe artifacts from the portrait relighting algorithm in Chapter 2 when relighting with environment maps that contain high-frequency illumination. We believe the method introduced in this Chapter can provide better training data and significantly improve many of these methods in the future.

3.3 Model

An OLAT scan of a subject captured by a light stage consists of n images, where each image is lit by a single light in the stage. The conventional way to relight the captured subject with an arbitrary light direction is to linearly blend the images captured under nearby lights in the OLAT scan. As shown in Fig. 4.1, this often results in “ghosting” artifacts on shadows and highlights. The goal of this work is to use machine learning instead of simple linear interpolation to produce higher-quality results. Our model takes as input a query light direction ℓ and a complete OLAT scan consisting of a set of paired images and light directions $\{\mathbf{I}_i, \ell_i\}$, and uses a deep neural network Φ to obtain the predicted image $\mathbf{I}$,

$$\mathbf{I}(\ell) = \Phi(\{\mathbf{I}_i, \ell_i\}_{i=1}^n, \ell). \quad (3.1)$$

This formalization is broad enough to describe some prior works on learning-based relighting [130, 69]. While these methods usually operate by training a U-Net [93] to map from a *sparse* set of input images to an output image, we focus on producing as high-quality as possible rendering results given the *complete* OLAT scan. However, feeding all the captured images into a conventional CNN network is not tractable in terms of speed or memory requirements. In addition, this naive approach seems somewhat excessive for practical applications involving human faces. While complex translucency and interreflection may require multiple lights to

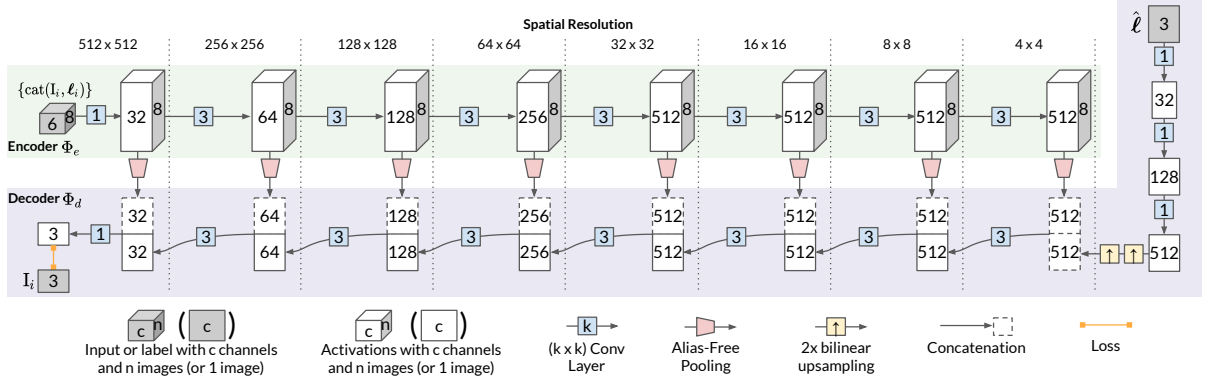


Figure 3.2. A visualization of our model architecture. The encoder of our model $\Phi_e(\cdot)$ takes as input a concatenation of the nearby OLAT images in the active set and their light directions, which are processed by a series of stride-2 conv layers. The resulting encoded activations of these 8 images at each level are then combined using the alias-free pooling described in Section 3.3.2, and skip-connected to the decoder. The decoder $\Phi_d(\cdot)$ takes as input the query light direction ℓ , processes it with fully connected layers and then upsamples it (along with the skip-connected encoder activations), and decodes the image using a series of stride-2 transposed conv layers. Whether or not a conv or transposed conv changes resolution is indicated by whether or not its edge spans two spatial scales.

reproduce, it is unlikely that *all* images in the OLAT scan are necessary to reconstruct the image for any particular query light direction, especially given that barycentric interpolation requires only three nearby lights to produce a somewhat plausible rendering. Our work attempts to find an effective and tractable compromise between these two extremes, in which the power of deep neural networks is combined with the efficiency and generality of nearest-neighbor approaches. This is accomplished by a linear blending approach that (like barycentric blending) ensures the output rendering is a smooth function of the input, where the blending is performed on the activations of a neural network’s encoding of our input images instead of on the raw pixel intensities of the input images.

Our complete network structure is shown in Fig. 3.2. Given a query light direction ℓ , we identify the k captured images in the OLAT scan whose corresponding light directions are nearby the query light direction, which we call *active set* $A(\ell)$. These OLAT images I_i and their corresponding light directions ℓ_i are then each independently processed in parallel by the encoder $\Phi_e(\cdot)$ of our convolutional neural network (or equivalently, they are processed as a single

“batch”), thereby producing a multi-scale set of internal neural network activations that describe all k images. After that, the set of k activations at each layer of the network are pooled into a single set of activations at each layer, which is performed using a weighted averaging where the weighting is a function of the query light and each input light $W(\ell, \ell_i)$. This weighted average is designed to remove the aliasing introduced by the nearest neighbor sampling in the active set selection stage. Together with the query light direction ℓ , these pooled feature maps are then fed into the decoder $\Phi_d(\cdot)$ by means of skip links from each level of the encoder, thereby producing the final predicted image $\mathbf{I}(\ell)$. Formally, our final image synthesis procedure is:

$$\mathbf{I}(\ell) = \Phi_d \left(\sum_{i \in A(\ell)} W(\ell, \ell_i) \Phi_e(\mathbf{I}_i, \ell_i), \ell \right). \quad (3.2)$$

This hybrid approach of nearest-neighbor selection and neural network processing allows us to learn a single neural network that produces high quality results, and generalizes well across query light directions and across subjects in our OLAT dataset.

Our active set construction approach is explained in Section 3.3.1, our alias-free pooling is explained in Section 3.3.2, the network architecture is described in Section 3.3.3, and our progressive training procedure is discussed in Section 3.3.4.

3.3.1 Active Set Selection

Light stages are conventionally constructed by placing lights on a regular hexagonal tessellation of a sphere (with some “holes” for cameras and other practical concerns), as shown in Fig. 3.3. As discussed, at test time our model works by identifying the OLAT images and lights that are nearest to the desired query light direction, and averaging their neural activations. But this natural approach, when combined with the regularity of the sampling of lights in the light stage, presents a number of problems for training our model. First, we can only supervise our super-resolution model using “virtual” lights that exactly coincide with the real lights of the light stage, as these are the only light directions for which we have ground-truth images (this will also

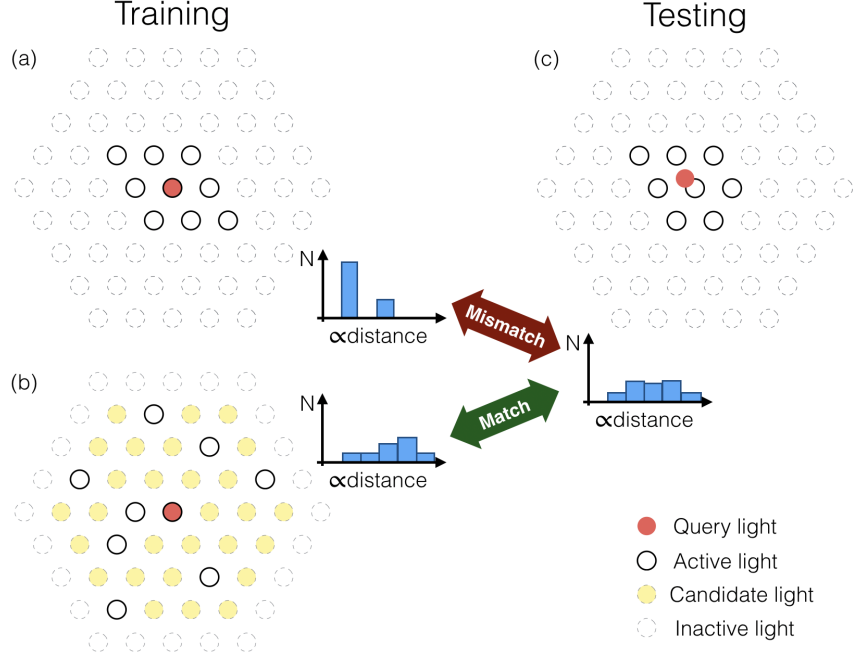


Figure 3.3. The OLAT images taken from a light stage have a uniform hexagonal pattern, which means that the distances between each light and its nearest neighbors is highly regular (a). In contrast, at test time we want to synthesize images corresponding to unseen light directions that do not lie on this hexagonal grid, and whose neighboring distances will therefore be irregular (c). During training we therefore sample a random subset of nearest neighbors for use in the active set of our model (b), which forces the network to adapt to challenging and irregular distributions of neighbor-distances that better match those that will be seen at test time.

be a problem when evaluating our model, as will be discussed in Sec. 3.4). Second, this regular hexagonal sampling means that, for any given light in the stage, the distances between it and its neighbors will always exhibit a highly regular pattern (Fig. 3.3a). For example, the 6 nearest neighbors of every point on a hexagonal tiling are guaranteed to have exactly the same distance to that point. In contrast, at test time we would like to be able to produce renderings for query light directions that correspond to arbitrary points on the sphere, and those points will likely have irregular distributions of neighboring lights (Fig. 3.3c). This represents a significant deviation between our training data and our test data, and as such we should expect poor generalization at test time if we were to naively train on highly-regular sets of nearest neighbors.

To address this issue, we adopt a different technique for sampling neighbors for use in our active set than what is used during test time. For each training iteration, we first identify a larger

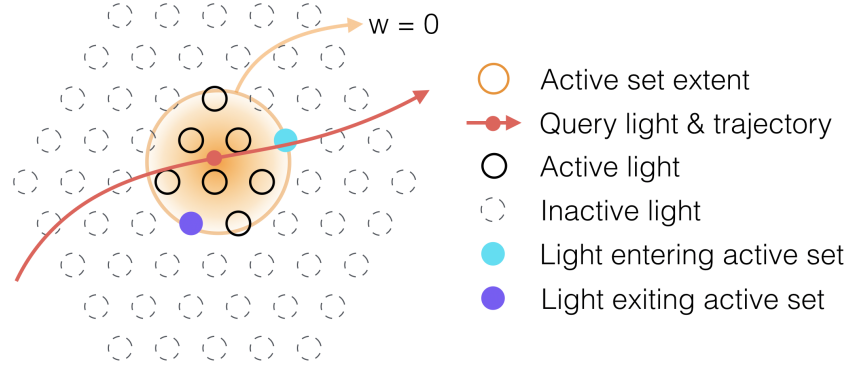


Figure 3.4. Varying the query light direction will cause OLAT images to leave and enter the active set of our model, which introduces aliasing that, if unaddressed, results in jarring temporal artifacts in our renderings. To address this, we use an “alias-free pooling” technique to ensure that the network activations of each OLAT image are averaged in a way that suppresses this aliasing. We use a weighted average where the weights are smooth, and are exactly zero at the point where lights enter and leave the active set.

set of m nearest neighbors near the query light (which in this case is identical to one of the real lights in the stage), and among them randomly select only $k < m$ neighbors to use in the active set (in practice, we use $m = 16$ and $k = 8$). As shown in Fig. 3.3b, this results in irregular neighbor sampling patterns during training, which simulates our test-time scenario wherein the query light is at a variety of locations in between the real input light sources. This approach shares a similar motivation as that of dropout [108] in neural networks, in which network activations are randomly set to 0 during training to prevent overfitting. Here we instead randomly remove input images, which also has the effect of preventing the model from overfitting to the hexagonal pattern of the light stage while training our network, by forcing it to operate on more varied inputs. Notice that the query light itself is included in the candidate set, to reflect the fact that during test-time the “virtual” query light may be next to a real light source. As we will show in Sec. 3.4, this active set selection approach results in a learned model whose synthesized shadows move more smoothly and at a more regular rate than is achieved with a naive nearest-neighbor sampling approach.

3.3.2 Alias-Free Pooling

A critical component in our model is the design of the skip links from each level of the encoder of our model to its corresponding level in the decoder. This model component is responsible for the network activations corresponding to the 8 images in our active set and reducing them to one set of activations corresponding to a single output, which will then be decoded into an image. This requires a pooling operator for these 8 images. This pooling operator must be permutation-invariant, as the images in our active set may correspond to any OLAT light direction and may be presented in any order. Standard permutation-invariant pooling operators, such as average-pooling or max-pooling, are not sufficient for our case, because they do not suppress *aliasing*. As the query light direction moves across the sphere, images will enter and leave the active set of our model, which will cause the network activations within our encoder to change suddenly (see Fig. 3.4). If we use simple average-pooling or max-pooling, the activations in our decoder will also vary abruptly, resulting in unrealistic flickering artifacts or temporal instability in our output renderings as the light direction varies. In other words, the point sampled signal should go through an effective prefiltering process in order to suppress the artifacts.

The root cause of this problem is that our active set is an aliased observation of the input images, and average- or max-pooling allows this aliasing to persist. We therefore introduce a technique for alias-free pooling to address this issue. We use a weighted average as our pooling operator where the weight of each item in our active set is a continuous function of the query light direction, and where the weight of each item is guaranteed to be zero at the moment it enters or leaves the active set. We define our weighting function between the query light direction ℓ and each OLAT light direction ℓ_i as follows:

$$\begin{aligned}\tilde{W}(\ell, \ell_i) &= \max \left(0, e^{s(\ell \cdot \ell_i - 1)} - \min_{j \in A(\ell)} e^{s(\ell \cdot \ell_j - 1)} \right), \\ W(\ell, \ell_i) &= \frac{\tilde{W}(\ell, \ell_i)}{\sum_j \tilde{W}(\ell, \ell_j)},\end{aligned}\tag{3.3}$$

where s is a learnable parameter that adjusts the decay of the weight with respect to the distance and each ℓ is a normalized vector in 3D space. During training, parameter s will be automatically adjusted to balance between selecting the nearest neighbor ($s = +\infty$) and taking an unweighted average of all neighbors ($s = 0$).

Our weighting function is an offset spherical Gaussian, similar to the normalized Gaussian distance between the query light’s Cartesian coordinates and those of the other lights in our active set, but where we have subtracted out the unnormalized weight corresponding to the most distant light in the active set (and clipped the resulting weights at 0). This adaptive truncation is necessary because the lights in the light stage may be spaced irregularly (due to holes in the stage for cameras or other reasons), which means that a fixed truncation may be too aggressive in setting weights to zero in regions where lights are sampled less frequently. We instead leverage the fact that when a light exits the active set, a new light will enter it at exactly the same time with exactly the same distance to the query light. This allows us to truncate our Gaussian weights using the maximum distance in the active set, which ensures that lights have a weight of zero as they leave or enter the active set. This results in renderings that change smoothly as we move the query light direction.

3.3.3 Network Architecture

The remaining components of our model consist of the conventional building blocks used in constructing convolutional neural networks, and can be seen in Fig. 3.2. The encoder of our network consists of 3×3 convolutional neural network blocks (with a stride of 2 so as to reduce resolution by half), each of which is followed by group normalization [127] and a PReLU [37] activation function. The number of hidden units for each layer begins at 32 and doubles after each layer, but is clipped at 512. The input to our encoder is a set of 8 RGB input images corresponding to the nearby OLAT images in our active set, each of which has been concatenated with the xyz coordinate of its source light (tiled to every pixel) giving us 8 6-channel input images.

These images are processed along the “batch” dimension of our network, and so are treated identically at each level of the encoder. These 8 images are then pooled down to a single “image” (*i.e.*, a single batch) of activations using the alias-free pooling approach of Section 3.3.2, each of which is concatenated onto the internal activations of the network’s decoder.

The decoder of the network begins with a series of fully-connected (aka “dense”) neural network blocks that take as input the query light direction ℓ , each of which is followed by instance normalization [117] and a PReLU activation function. These activations are then upsampled to 4×4 and used as the basis of our decoder. Each layer of the decoder consists of a 3×3 transposed convolutional neural network block (with a stride of 2 so as to double resolution) which is again followed by group normalization and a PReLU activation function. The input to each layer’s conv block is a concatenation of the upsampled activations from the previous decoder level, with the pooled activations from the encoder that have been “skip” connected from the same spatial scale. The final activation function before any output image is produced is a sigmoid function, as our images are normalized to $[0, 1]$. Because our network is fully convolutional [64], it can be evaluated on images of arbitrary resolution, with GPU memory being the only limiting factor. We train on 512×512 resolution images for the sake of speed, and evaluate and test on 1024×1024 resolution images to maximize image quality.

3.3.4 Loss Functions and Training Strategy

We supervise the training of our model using an L_1 loss on pixel intensities. Formally, our loss function is:

$$\mathcal{L}_d = \sum_i \|\mathbf{M} \odot (\mathbf{I}_i - \mathbf{I}(\ell_i))\|_1, \quad (3.4)$$

where $\mathbf{I}_i$ is the ground truth image under light i , and $\mathbf{I}(\ell_i)$ is our prediction. When computing the loss over the image, we use a precomputed binary image $\mathbf{M}$ to mask out pixels that are known to belong to the background of the subject.

During training, we construct each training data instance by randomly selecting a human

subject in our training dataset and then randomly selecting one OLAT light direction i . The image corresponding to that light $\mathbf{I}_i$ will be used as the ground-truth image our model will attempt to reconstruct, and the “query” light direction for our model will be the light corresponding to that image ℓ_i . We then identify a set of 8 neighboring images/lights to include in our active set using the selection procedure described in Section 3.3.1. Our only data augmentation is a randomly-positioned 512×512 crop of all images in each batch.

Progressive training has been found to be effective for accelerating and stabilizing the training of GANs for high-resolution image synthesis [46], and though our model is not a GAN (but is instead a convolutional encoder-decoder architecture with skip connections) we found it to also benefit from a progressive training strategy. We first inject downsampled image inputs directly into a coarse layer of our encoder and supervise training by imposing a reconstruction loss at a coarse layer of our decoder, resulting in a shallower model that is easier to train. As training proceeds, we add additional convolutional layers to the encoder and decoder, thereby gradually increasing the resolution of our model until we arrive at the complete network and the full image resolution. In total, we train our network for 200,000 iterations, using 8 NVIDIA V100 GPUs, which takes approximately 10 hours.

Our model is implemented in Tensorflow [1] and trained using Adam [53] with a batch size of 1 (the “batch” dimension of our tensors is used to represent the 8 images in our active set), a learning rate of 10^{-3} , and default hyperparameter settings ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$).

3.4 Evaluation

We use the OLAT portrait dataset from [109], which contains 22 subjects with multiple facial expressions captured using a light stage and a 7-camera system. The light stage consists of 302 LEDs uniformly distributed on a spherical dome, and capturing a subject takes roughly 6 seconds. Each capture process produces an OLAT scan of a specific facial expression on each camera, which consists of 302 images, and we treat the OLAT scans from different cameras

as independent OLAT scans. Because the subject is asked to stay still (and an optical flow algorithm [125] is applied to correct the small movements) the captured 302 images in each OLAT are aligned and only differ in lighting directions. We manually select 4 OLAT scans with a mixture of subjects and views for use as our validation set, and choose another 16 OLAT scans with good coverage of gender and diverse skin tones for use as training data. Our 16 training datasets only covers 5 of 7 cameras, and the remaining 2 are covered by the validation data. We train our network using all lights from our OLAT data in a canonical global lighting coordinate frame, which allows us to train a single network for all viewpoints in our training data. We train one single model for all subjects in our training dataset, which we found to match the performance of training an individual model for each subject.

Empirically evaluating our model presents a significant challenge: our model is attempting to super-resolve an undersampled scan from a light stage, which means that the only ground-truth that is available for benchmarking is *also* undersampled. In other words, the goal of our model is to accurately synthesize images that correspond to virtual lights in between the real lights of the stage — but we do not have ground-truth images that correspond to those virtual lights. In addition, the model also needs to generalize to an unseen view and subject. For these reasons, qualitative results (figures, videos) are preferred, and we encourage readers to view our figures and the online video. In the quantitative results presented here, we use held-out real images lit by real lights on our light stage as a validation set. When evaluating one of these validation images, we do not use the active-set selection technique of Section 3.3.1, and instead just sample the $k = 8$ nearest neighbors (excluding the validation image itself from the input). Holding out the validation image from the inputs is critical, as otherwise a model could simply reproduce the input image as an error-free output. This held-out validation approach is not ideal, as all such evaluations will follow the same regular sampling pattern of our light stage. This evaluation task is therefore more biased than the real task of predicting images away from the sampling pattern of the light stage.

Selecting an appropriate metric for measuring image reconstruction accuracy for our

Table 3.1. Here we benchmark our model against prior work and ablations of our model on our validation dataset. We report the arithmetic mean of each metric across the validation set. The top three results of each metric are highlighted in red, orange, yellow, respectively. While “Ours w/naive neighbors” has the lowest error according to this evaluation, “Our model” performs better in our *real* test-time scenario where the synthesized light does not lie in a regular hexagonal grid (see text and Fig. 3.5 for details).

Algorithm	RMSE	H^1	DSSIM	E-LPIPS
Our model	0.0160	0.0203	0.0331	0.00466
Ours w/naive neighbors	0.0156	0.0199	0.0322	0.00449
Ours w/avg-pooling	0.0203	0.0241	0.0413	0.00579
Linear blending	0.0191	0.0232	0.0366	0.00503
Fuchs <i>et al.</i> [27]	0.0195	0.0258	0.0382	0.00485
Photometric stereo	0.0284	0.0362	0.0968	0.00895
Xu <i>et al.</i> [130]				
w/ 8 optimal lights	0.0410	0.0437	0.1262	0.01666
w/ adaptive input	0.0259	0.0291	0.1156	0.00916
Meka <i>et al.</i> [69]	0.0505	0.0561	0.1308	0.01482

task is not straightforward. Conventional image interpolation techniques often result in ghosting artifacts or duplicated highlights, which are perceptually salient but often not penalized heavily by traditional image metrics such as per-pixel RMSE. We therefore evaluate image quality using multiple image metrics: RMSE, the Sobolev H^1 norm [78], DSSIM [121], and E-LPIPS [51]. RMSE measures pixel-wise error, the H^1 norm emphasizes image gradient error, while DSSIM and E-LPIPS approximate an overall perceptual difference between the predicted image and the ground truth. Still, images and videos are preferred for comparison.

3.4.1 Ablation Study

We first evaluate against ablated versions of our model, with results shown in Tab. 3.1. In the “Ours w/naive neighbors” ablation we use the $k = 8$ nearest neighbors in our active set during training. This setup leads to a match between our training and validation data, which results in better numerical performance (as shown in Tab. 3.1) but also significant overfitting: this apparent improvement in performance is misleading, because the validation set of our dataset

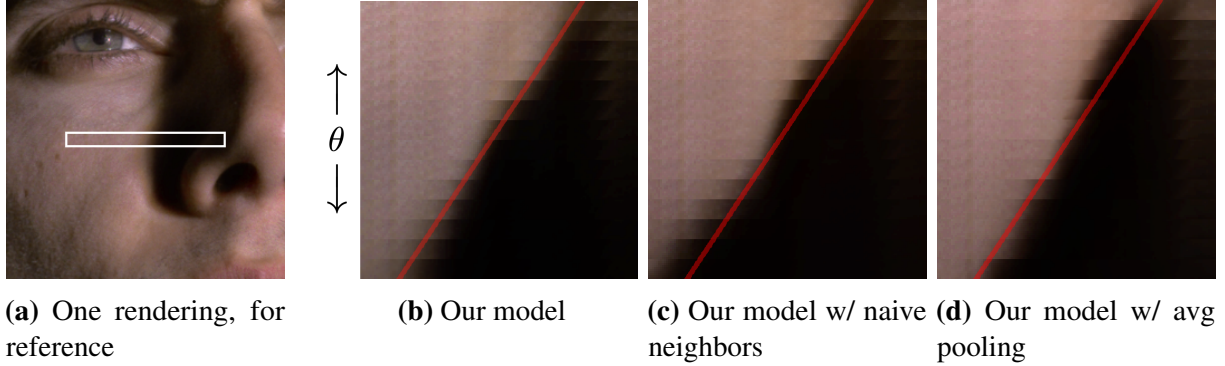


Figure 3.5. A visualization of how our learned model synthesizes renderings in which shadows move smoothly as a function of light direction. In (a) we show a rendering from our model for some virtual light ℓ with a horizontal angle of θ , and highlight one image strip that includes a horizontal cast shadow. In (b) we repeatedly query our model with θ values that should induce a linear horizontal translation of the shadow boundary in the image plane, and by stacking these image strips we can see this linear trend emerge (highlighted in red). In (c) and (d) we do the same for ablations of our model that do not have our active-set random selection procedure nor our alias-free pooling, and we see that the resulting shadow boundary does not vary smoothly or linearly.

has the same overly-regular sampling as the training set. During our *real* test-time scenario in which we synthesize with lights that do not lie on the regular hexagonal grid of our light stage, we see this ablated model generalizes poorly. In Fig. 3.5 we visualize the output of our model and ablations of our model as a function of the query light direction. We see that our model is able to synthesize a cast shadow that is a smooth linear function in the image plane of the angle of the query light (after accounting for foreshortening, *etc*). Ablations of our technique do not reproduce this linearly-varying shadow, due to the aliasing and overfitting problems described earlier.

In the “Ours w/avg-pooling” ablation we replace the alias-free pooling of our model with simple average pooling. As shown in Tab 3.1, ablating this component reduces performance. But more importantly, ablating this component also causes flickering during our *real* test-time scenario in which we smoothly vary our light source, and this is not reflected in our quantitative evaluation. Because average pooling assigns a non-zero weight to images as they enter and exit our active set, renderings from this model will contain significant temporal instability.

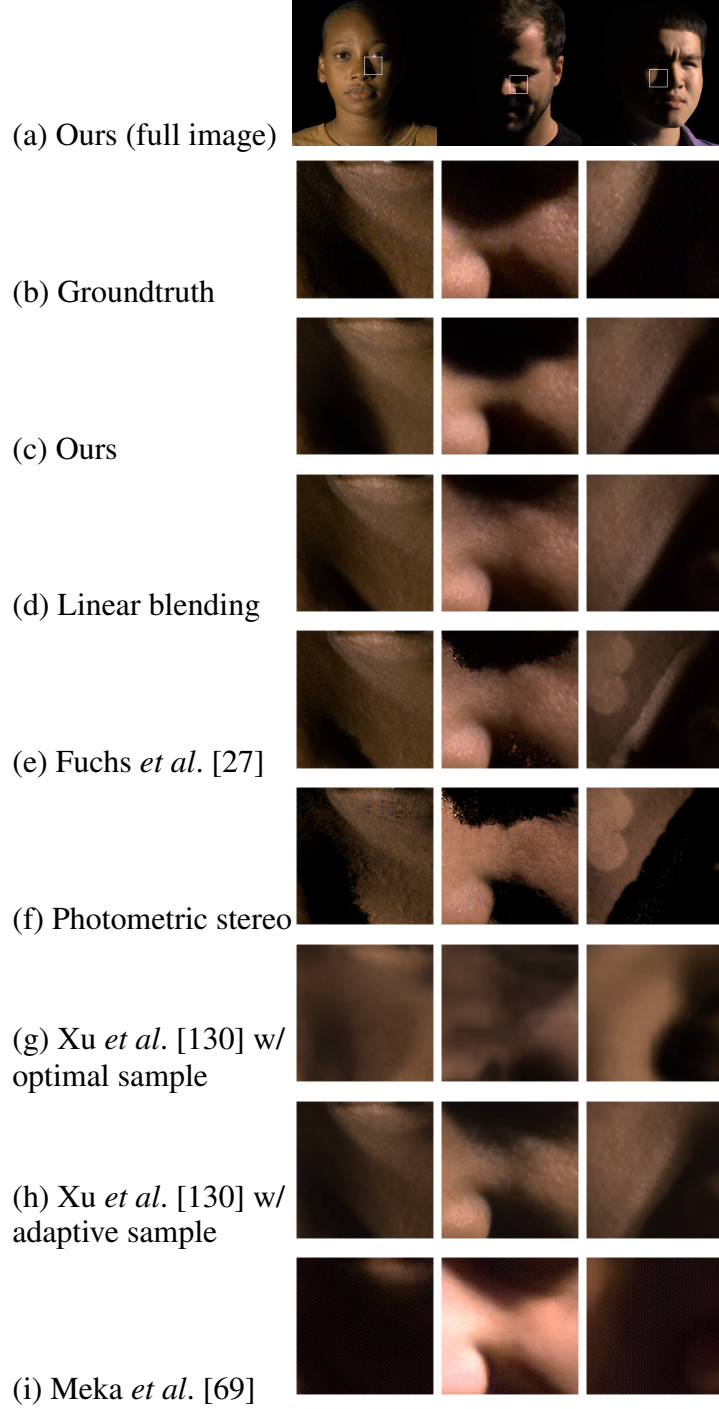


Figure 3.6. Here we present a qualitative comparison between our method and other light interpolation algorithms. Traditional methods (linear blending, Fuchs *et al.* [27], photometric stereo) retain detail but suffer from ghosting artifacts in shadowed regions. Results from Xu *et al.* [130] and Meka *et al.* [69] exhibit significant oversmoothing and brightness changes. Our method retains details and synthesizes shadows that resemble the ground truth.

3.4.2 Related Work Comparison

We compare our results against related approaches that are capable of solving the relighting problem. The “Linear blending” baseline in Tab. 3.1 produces competitive results, despite being a very simple algorithm: we simply blend the input images of our light stage according to our alias-free weights. Because linear blending directly interpolates aligned pixel values, it is often able to retain accurate high frequency details in the flat region, and this strategy works well for minimizing our error metrics. However, linear blending produces significant ghosting artifacts in shadows and highlights, as shown in Fig. 3.6. Though these errors are easy to detect visually, they appear to be hard to measure empirically.

We evaluate against the layer-based technique of Fuchs *et al.* [27] by decomposing an OLAT into diffuse, specular, and visibility layers, and interpolating the illumination individually for each layer. Although the method works well on specular objects as shown in the original paper, it performs less well on OLATs of human subjects, as shown in Tab. 3.1. This appears to be due to the complex specularities on human skin not being tracked accurately by the optical flow algorithm of Fuchs *et al.* [27]. Additionally, the interpolation of the visibility layer sometimes contains artifacts, which results in cast shadows being predicted incorrectly. That being said, the algorithm results in fewer ghosting artifacts than the linear blending algorithm, as shown in Fig. 3.6 and as reflected by the E-LPIPS metric.

Using the layer decomposition produced by Fuchs *et al.* [27], we additionally perform photometric stereo on the OLAT data by simple linear regression to estimate a per-pixel albedo image and normal map. Using this normal map and albedo image we can then use Lambertian reflectance to render a new diffuse image corresponding to the query light direction, which we add to the specular layer from [27] to produce our final rendering. As shown in Tab. 3.1, this approach underperforms that of Fuchs *et al.* [27], likely due to the reflectance of human faces being non-Lambertian. Additionally, the scattering effect of human hair is poorly modeled in terms of a per-pixel albedo and normal vector. These limiting assumptions result in overly

sharpened and incorrect shadow predictions, as shown in Fig. 3.6. In contrast to this photometric stereo approach and the layer-based approach of Fuchs *et al.* [27], our model does not attempt to factorize the human subject into a predefined reflectance model wherein interpolation can be explicitly performed. Our model is instead trained to identify a latent vector space of network activations in which naive linear interpolation results in accurate non-linearly interpolated images, which results in more accurate renderings.

The technique of Xu *et al.* [130] (retrained on our training data) represents another possible candidate for addressing our problem. This technique does not natively solve our problem. In order to find the optimal lighting directions for relighting, it requires as input *all* 302 high-resolution images in each OLAT scan in the first step, which significantly exceeds the memory constraints of modern GPUs. To address this, we first jointly train the Sample-Net and the Relight-Net on our images (downsampled by a factor of $4\times$ due to memory constraints) to identify 8 optimal directions from the 302 total directions of the light stage. Using those 8 optimal directions, we then retrain the Relight-Net using the full-resolution images from our training data, as prescribed in Xu *et al.* [130]. Table 3.1 shows that this approach works poorly on our task. This may be because this technique is built around 8 fixed input images and is naturally disadvantaged compared to our approach, which is able to use any of the 302 light stage images as input. We therefore also evaluate a variant of Xu *et al.* [130] where we use the same active-set selection approach used by our model to select the images used to train their Relight-Net. By using our active-set selection approach (Sec. 3.3.1) this baseline is able to better reason about local information, which improves performance as shown in Tab. 3.1. However, this baseline still results in flickering artifacts when rendering with moving lights, because (unlike our approach) it is sensitive to the aliasing induced when images leave and enter the active set.

We also evaluate Deep Reflectance Fields [69] for our task, which is also outperformed by our model. This is likely because their model is specifically designed for fast and approximate video relighting and uses only two images as input, while our model has access to the entire OLAT scan and is designed to prioritize high-quality rendering.

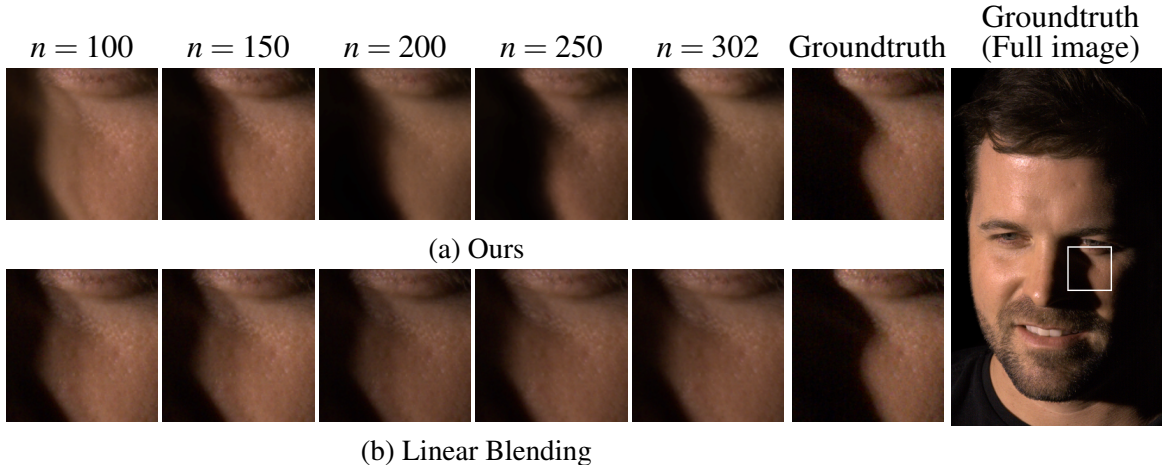


Figure 3.7. Here we compare the performance of our model against linear blending as we reduce n , the number of lights in our light stage. As we decrease the number of available lights from $n = 302$ to $n = 100$, the quality of our model’s rendered shadow degrades slowly. Linear blending, in contrast, is unable to produce an accurate rendering even with access to all lights.

3.4.3 Light Stage Subsampling

An interesting question in light transport acquisition is how many images (light samples) are needed to reconstruct the full light transport function. To address this question, we present an experiment in which we remove some lights from our training set and use only this subsampled data during training and inference. We reduce the number of lights on the light stage n (while maintaining a uniform distribution on the sphere) to $[250, 200, 150, 100]$, while also changing the number of candidates m and the active set size k to $[14, 12, 10, 8]$ and $[7, 6, 5, 4]$ respectively. Image quality on the complete validation dataset (with all 302 lights) as a function of the number of subsampled training/input lights is shown in Fig. 3.8. As expected, relighting quality decreases as we remove the lights, but we see that the rendering quality of our method decreases more slowly than that of linear blending. This can also be observed in Fig. 3.7, where we present relit renderings using these subsampled light stages. We see that removing lights reduces accuracy compared to the ground truth, but that our synthesized shadows remain relatively sharp: ghosting artifacts only appear when $n = 100$. In comparison, linear blending produces ghosting artifacts near shadow boundaries for all values of n . During test time, our model can also produce accurate

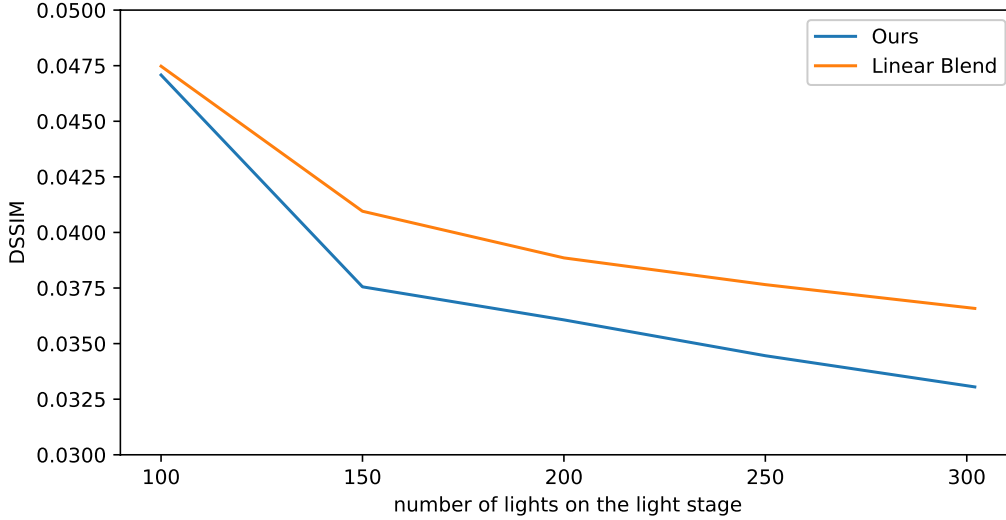


Figure 3.8. The image quality of relighting algorithms will gradually reduce as we remove lights from the light stage. However, our algorithm is able to retain the image quality to a greater extent with fewer lights compared to naive linear blending.

shadows and sharp highlights.

3.5 Continuous High-Frequency Relighting

A key benefit of our method is the ability to “super-resolve” an OLAT scan with virtual lights at a higher resolution than the original light stage data, thereby enabling continuous high-frequency relighting with an essentially continuous lighting distribution (or equivalently, with a light stage whose sampling frequency is unbounded). In this section, we present three applications of this idea.

Precise Directional Light Relighting

Traditional image-based relighting methods produce accurate results near the observed lights of the stage, but may introduce ghosting effects or inaccurate shadows when no observed light is nearby. In Fig. 3.9 we try to interpolate the image between two lights on the stage. As shown in the second and the third row, linear blending or Xu *et al.* [130] with adaptive sampling does not produce realistic results and always contains multiple superposed shadows or highlights. The shadows produced by Meka *et al.* [69] are sharp, but are not moving smoothly when the

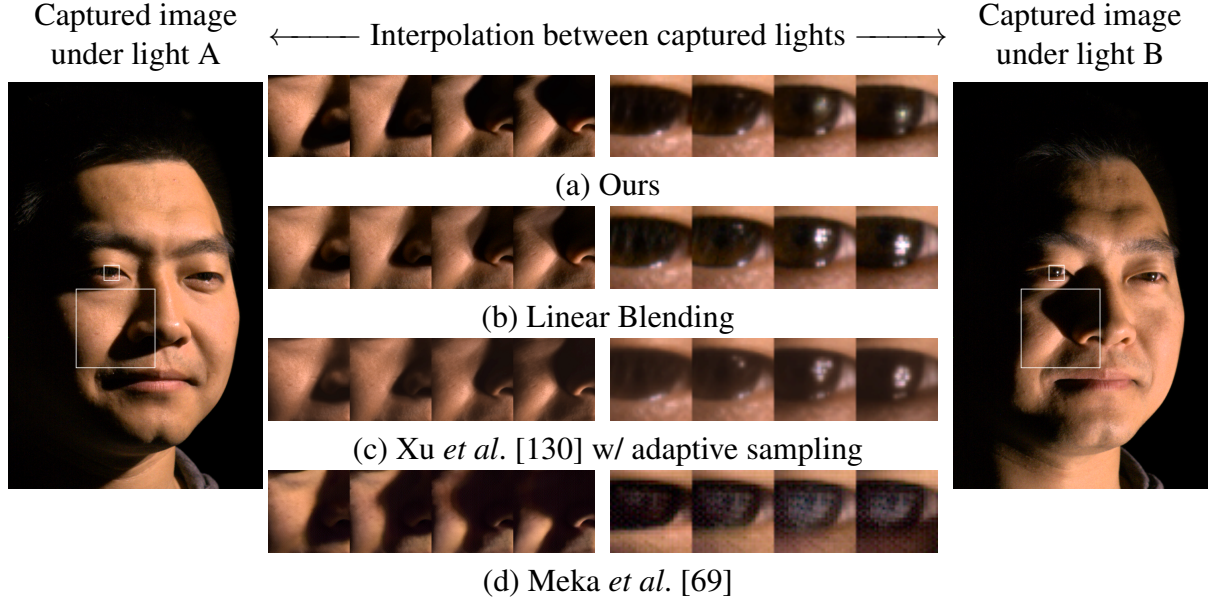


Figure 3.9. Here we use produce interpolated images corresponding to “virtual” lights between two real lights of the light stage. Our model (a) produces renderings where sharp shadows and accurate highlights move realistically. Linear blending (b) and Xu *et al.* [130] with adaptive sampling result in ghosting artifacts and duplicated highlights. The results from Meka *et al.* [69] contain blurry highlights and shadows with unrealistic motion.

light moves. In contrast, our method is able to produce sharp and realistic images for arbitrary light directions: highlights and cast shadows move smoothly as we change the light direction, and our results have comparable sharpness to the (non-interpolated) groundtruth images that are available.

High Frequency Environment Relighting

OLAT scans captured from a light stage can be linearly blended to reproduce images that appear to have been captured under a specific environment. The pixel values of the environment map are usually distributed to the nearest or neighboring lights on the light stage for blending. This traditional approach may cause ghosting artifacts in shadows and specularities, due to the finite sampling of light directions on the light stage. Although this ghosting is hardly noticeable when the lighting is low-frequency, it can be significant when the environment contains high frequency lighting, such as the sun in the sky. These ghosting artifacts can be ameliorated by

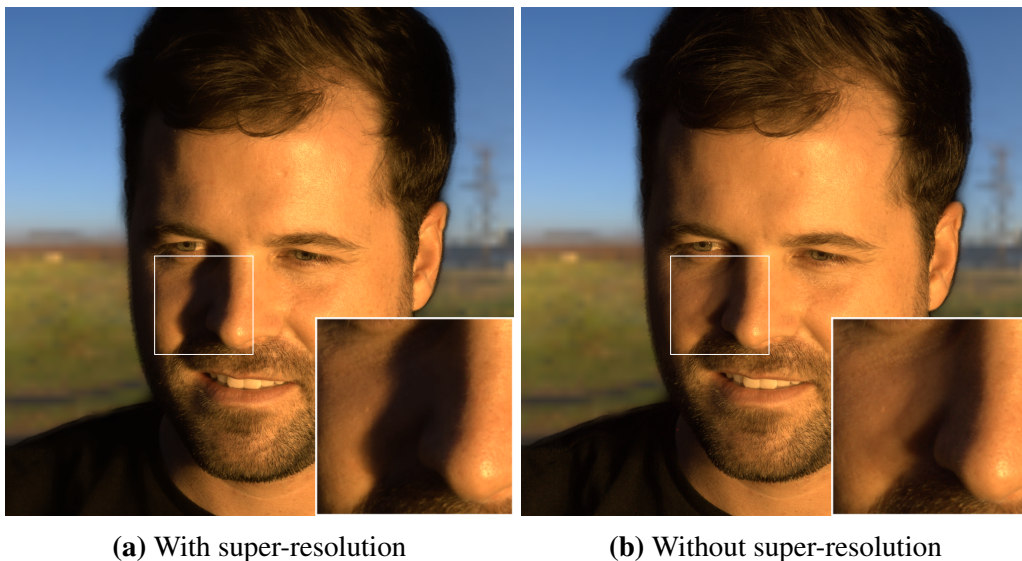


Figure 3.10. Our model (a) is able to produce accurate relighting results under high-frequency environments by super-resolving the light stage before performing image-based relighting [19]. Using the light stage data as-provided (b) results in ghosting.

using our model. Given an environment map, our algorithm can predict the image corresponding to the light direction of each pixel in the environment map. By taking a linear combination of all such images (weighted by their pixel values and solid angles), we are able to produce a rendering that matches the sampling resolution of the environment map. As shown in Fig. 3.10, this approach produces images with sharp shadows and minimal ghosting when given a high-frequency environment, while linear blending does not. In this example, we use an environment resolution of 256×128 , which corresponds to a super-resolved light stage with 32,768 lights.

We now analyze the relationship between the image quality gain from our model and the frequency of the lighting. Specifically, we evaluate for which environments, and at what frequencies, our algorithm will be required for accurate rendering, and conversely how our model performs in low-frequency lighting environments where previous solutions are adequate. For this purpose, we use one OLAT scan, and render it under 380 high quality indoor and outdoor environment maps (environments downloaded from hdrihaven.com) using both our model and linear blending. We then measure the image quality gain from our model by computing the DSSIM value between our rendering and that from linear blending. We measure the frequency

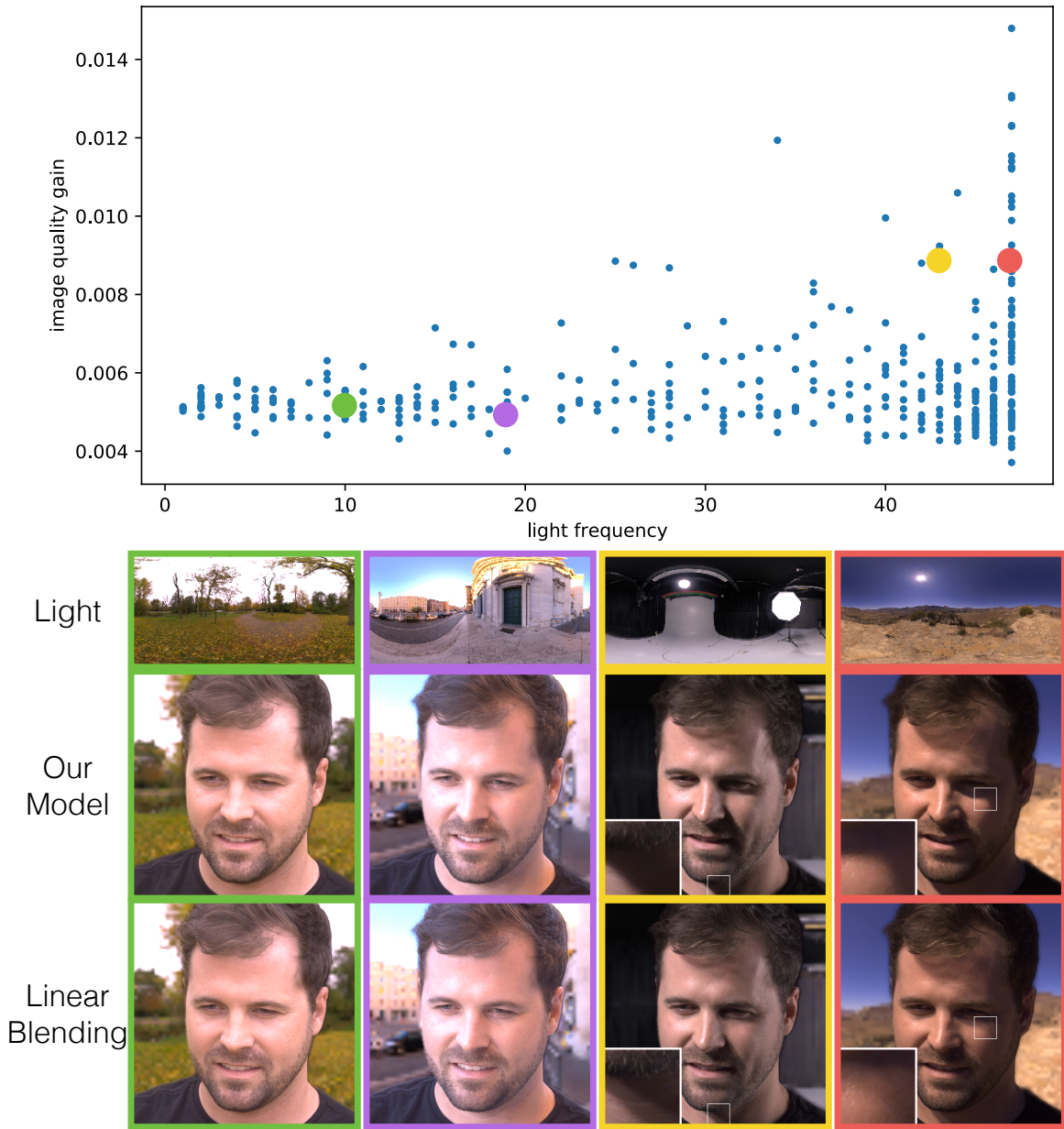


Figure 3.11. In the top figure, each blue dot represents a lighting environment. We render a portrait under this environment using both linear-blending and our method, and measure the image difference using SSIM to evaluate the quality gain of our algorithm. The image quality improvement produced by our model becomes more apparent when the environment map has more high-frequency variation. In the bottom figure, we compare the rendered images using our model and linear blending under environment maps with different frequencies. Our model produces similar results to linear blending when the lighting variation is low frequency (left columns). As the lighting variation becomes higher frequency, our model produces better renderings with fewer artifacts and sharper shadows (right columns).

of the environmental lighting by decomposing it into spherical harmonics (up to degree 50), and finding the degree below which 90% of the energy can be recovered.

As shown in Fig. 3.11, the benefit of using our model becomes larger when the frequency of the environment increases. For low-frequency light (up to degree 15 spherical harmonics), our model produces almost identical results compared to the traditional linear blending method. This is a desired property, showing that our method reduces gracefully to linear blending for low frequency lighting, and thus produces high quality results for any low or high-frequency environment. As the frequency of the lighting becomes higher, the renderings of our model contain sharper and more accurate shadows without ghosting artifacts. Note that there is some variation among the environment maps as expected; even a very high-frequency environment could coincidentally have its brightest lights aligned with one of the light in the light stage, leading to low error in linear blending and comparable results to our method. Nevertheless, the trend is clear in Fig. 3.11 with many high-frequency environments requiring our algorithm for lighting super-resolution.

According to the plot, we conclude that our model is necessary when the light frequency is equal or larger than about 20, which means more than $21^2 = 441$ basis functions are needed to recover the lighting. This number has the same order as the number of lights in the stage ($n = 302$). This observation agrees with intuition and frequency analysis. If the environment cannot be recovered using the limited lighting basis in the light stage, then light super-resolution is needed to generate new bases in order to accurately render the shadow and highlights.

Lighting Softness Control

Our model’s ability to render images under arbitrary light directions also allows us to control the softness of the shadow. Given a light direction, we can densely synthesize images corresponding to the light directions around it, and average those images together to produce a rendering with realistic soft shadows (the sampling radius of these lights determines the softness of the resulting shadow). As shown in Fig. 3.12, our model is able to synthesize realistic shadows

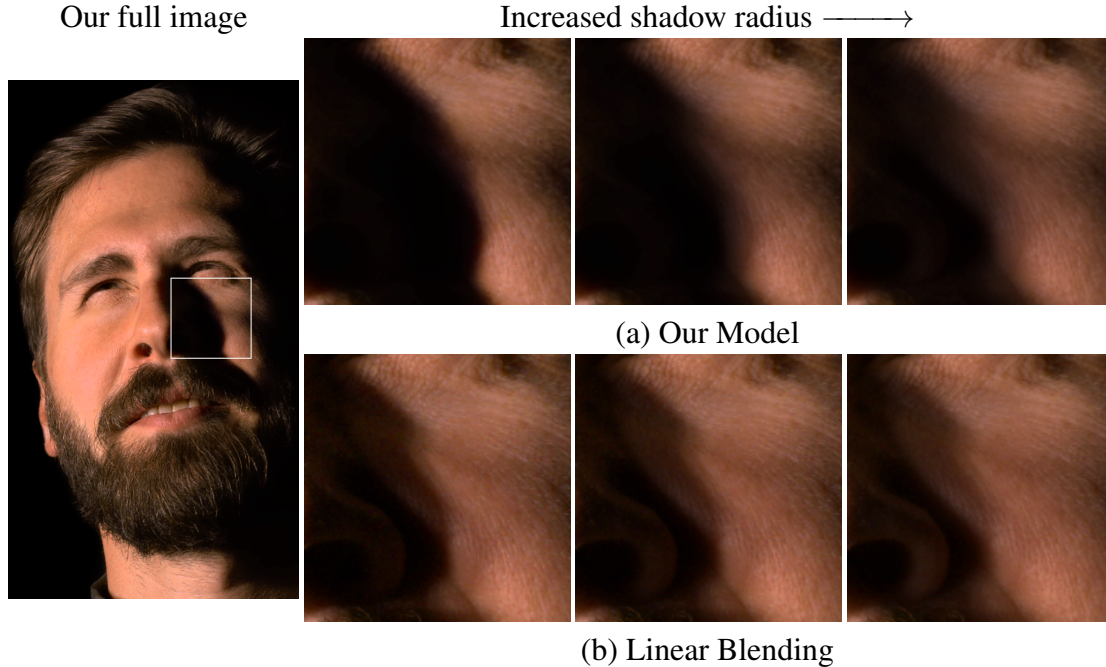


Figure 3.12. Soft shadows can be rendered by synthesizing and averaging images corresponding to directional light sources within some area on the sphere. Soft shadows rendered by our method (a) are more realistic and contain fewer ghosting artifacts than those rendered using linear blending (b).

with controllable softness, which is not possible using traditional linear blending methods.

3.6 Conclusions and Future Work

The light stage is a crucial tool for enabling the image-based relighting of human subjects in novel environments. But as we have demonstrated, light stage scans are undersampled with respect to the angle of incident light, which means that synthesizing virtual lights by simply combining images results in ghosting on shadows and specular highlights. We have presented a learning-based solution for super-resolving light stage scans, thereby allowing us to create a “virtual” light stage with a much higher angular lighting resolution, which allows us to render accurate shadows and highlights in high-frequency environment maps. Our network works by embedding input images from the light stage into a learned space where network activations can then be averaged, and decoding those activations according to some query light direction

to reconstruct an image. In constructing this model, we have identified two critical issues: an overly regular sampling pattern in light stage training data, and aliasing introduced when pooling activations of a set of nearest neighbors. These issues are addressed through our use of a dropout-like supersampling of neighbors in our active set, and our alias-free pooling technique. By combining ideas from conventional linear interpolation with the expressive power of deep neural networks, our model is able to produce renderings where shadows and highlights move smoothly as a function of the light direction.

This work is by no means the final word for the task of light stage super-resolution or image-based rendering. Approaches similar to ours could be applied to other general light transport acquisition problems, to other physical scanning setups, or to other kinds of objects besides human subjects. Though our network can work on inputs with different image resolutions, GPU memory has been a major bottleneck to apply our approach on images with much higher resolutions such as 4K resolution. A much more memory efficient approach for light-stage super-resolution is expected for production level usage in the visual effects industry. Though we exclusively pursue the one-light-at-a-time light stage scanning approach, alternative patterns where multiple lights are active simultaneously could be explored, which may enable a more sparse light stage design. Though the undersampling of the light stage is self-evident in our visualizations, it may be interesting to develop a formal theory of this undersampling with respect to materials and camera resolution, so as to understand what degree of undersampling can be tolerated in the limit. We have made a first step in this direction with the graph in Fig. 3.11. Finally, it would be interesting to extend our approach to enable the synthesis of novel viewpoints in addition to lighting directions. We believe that light stage super-resolution represents an exciting direction for future research, and has the potential to further decrease the time and resource constraints required for reproducing accurate high-frequency relighting effects.

This chapter is a reformatted version of the material as it appears in “Light stage super-resolution: continuous high-frequency relighting”, Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T. Barron, and Ravi

Ramamoorthi, ACM Transactions on Graphics (TOG) 39 (6), 2019 [111]. The dissertation author was the primary investigator and author of this paper.

Chapter 4

NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting

4.1 Introduction

Previous Chapters mainly focus on the problem of portrait relighting. Lighting is indeed a very important part for observing a portrait. However, human portraits also exhibit different appearances when viewed from a different camera angle. In this Chapter, we seek solutions to not only change the lighting but also the viewpoint of a portrait given some captured portrait.

Digitizing human portraits from natural portrait images and re-synthesizing novel images under new lighting and viewpoints is a long-standing graphics and vision problem with many applications like computational photography and video conferencing. Recently, NeRF [74] has initiated significant progress in neural rendering for photo-realistic image synthesis. Various radiance field extensions [107, 65, 7, 61] have been presented and many of the concurrent works [28, 63, 82] aim to address problems in human portrait capture.

However, previous NeRF-based portrait capture methods usually require capturing a large number of images and an expensive per-scene optimization process. Besides, most works consider only view synthesis tasks and do not support relighting. In general, a relightable portrait representation cannot simply be a radiance field that bakes in the original lighting condition; it instead requires explicitly disentangling and modeling lighting and portrait appearance in the image captures, which is a highly challenging problem.

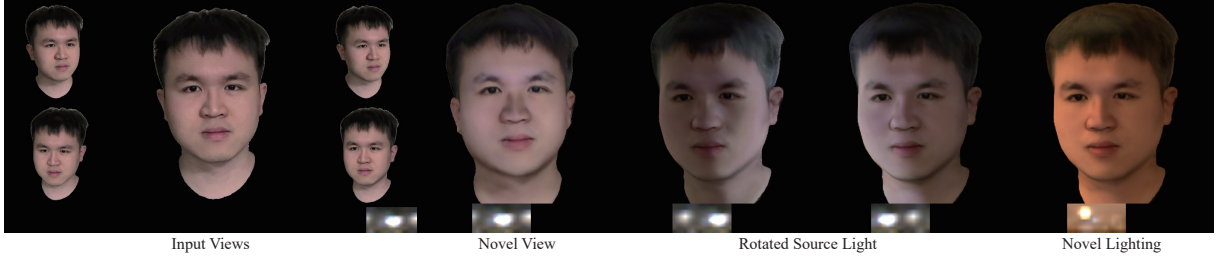


Figure 4.1. We introduce neural light-transport field (NeLF), which learns to infer the light-transport and volume density from a sparse set of input views. NeLF enables joint relighting and view synthesis of real portraits from only five input images.

In this Chapter, we propose a novel neural rendering approach that can jointly estimate the geometry and appearance of a human portrait and the lighting environment the human stands in, from only a sparse set of input images. We present a novel deep neural network that can generalize across scenes to regress a *neural light transport field*, i.e. NeLF, from portrait images. At an arbitrary 3D point, this neural light transport field outputs the volume density and light transport coefficients (unlike the view-dependent radiance in NeRF), which linearly explains the portrait appearance under distant illumination conditions represented by environment maps, similar to the classic light transport functions in image-based relighting [20] and precomputed radiance transfer [106, 79]. With this novel representation, our approach enables high-quality portrait relighting and view synthesis done simultaneously, as shown in Fig. 4.1.

Our approach incorporates classic light transport functions into neural volumetric rendering. In particular, we first use a UNet-like CNN as a feature extractor to convert each input portrait image to a neural feature map that encodes per-view pixel-wise portrait geometry and appearance. For an arbitrary 3D point, we fetch multi-view neural features from its projections in the feature maps and use MLPs to regress volume density and radiance from the features to enable differentiable ray marching. As opposed to earlier works [132, 120] that directly output radiance per shading point on marching rays, we propose to first regress a *light transport vector* – that can linearly compute the radiance under any novel lighting conditions – to enable relighting for neural volumetric rendering. In addition, we apply a sub-network to estimate the lighting as

environment maps from the bottleneck features of the CNN feature extractor, contributing to the disentanglement of portrait appearance and light effects in the original images. Our approach jointly estimates portrait geometry (as volume density), appearance (as light transport vectors) and lighting conditions (as environment maps) from portrait images, and can regress final ray colors under arbitrary novel viewpoints and lighting via differentiable ray marching.

Acquiring a real portrait dataset under different lighting conditions is a notoriously challenging task and traditionally requires a sophisticated light stage [20] that is not easily accessible for most researchers. In order to train our network with practical data, we instead utilize rendered images of human head models with different views and environment maps. We render our training and validation sets using the reconstructed 3D models of real human heads from a public dataset, FaceScape [131], leading to realistic renderings close to real portraits. In addition, we propose an effective domain adaptation module to enhance the generalizability of our network to real captured images. In particular, we apply additional CNN layers, appending to our CNN feature extractor, to regress the original input image sent to the feature extractor. We train this regression module using a large number of real portrait images in CelebAMask-HQ [58], which effectively regularizes our feature extractor along with our full network to understand diverse real portrait appearance. This regularization module effectively improves our rendering quality on real portrait images (compared with the rightmost image in Fig. 4.5).

We train our entire network from end to end with a final loss that combines the lighting estimation loss, domain regularization loss, and governing rendering losses with multiple combinations of various lighting and viewing conditions. This leads to our final generalizable network that can synthesize realistic portrait images under novel viewpoints and lighting conditions from only five input images. We demonstrate that, when evaluating on the rendered validation set, our approach can produce smooth and realistic relighting and view synthesis results that are very close to the ground truth; ours can qualitatively and quantitatively outperform baseline solutions that run state-of-the-art view synthesis [120] and relighting [109] techniques in a sequence (see Fig. 4.4). We also demonstrate that our approach can achieve photo-realistic renderings from

captured real portrait images, significantly better than the comparison methods (see Fig. 4.7).

We summarize our contributions as follows:

- a novel neural representation that models scene appearance as light transport functions and enables relighting for neural volumetric rendering (Sec. 4.3.1, Sec. 4.3.2);
- a domain adaptation module to enhance the generalizability of the network trained on rendered images (Sec. 4.3.4);
- realistic practical rendering results of joint relighting and view synthesis of real portraits from only five captured images (Sec. 4.4.4, Fig. 4.1, Fig. 4.7).

4.2 Related Work

4.2.1 Portrait Appearance

There has been extensive research work in capturing and modeling human portraits [10, 20, 50, 113, 102, 137]. Various morphable face models have been presented [113, 31, 124], mainly aiming to achieve facial animation and reenactment. Our focus is to capture human portraits and synthesize photo-realistic images. While generative models [33, 46, 47] can produce realistic images, they cannot be easily applied for capture and reconstruction, which aims to generate images of specific captured real human heads with given viewpoints and lighting conditions. While previous work can do so by reconstructing meshes with simple reflectance models from portrait images [4, 102], their rendering quality is often limited. We instead leverage the recent neural rendering techniques [112], leading to photo-realistic portrait renderings for joint view synthesis and relighting.

4.2.2 Relighting

Many image-based relighting papers have been presented [20, 67, 84, 91]; they leverage the linearity of light transport and introduce various techniques to acquire the light transport function, whose discrete form is a light transport matrix that consists of per-pixel light transport

vectors at a fixed viewpoint. Inspired by these prior works, we also model linear light transport functions; instead of predicting per-pixel coefficients, we predict light transport at a point in the 3D space, enabling both view synthesis and relighting.

Recently, deep learning techniques have been introduced to address the relighting problem [130]. Many deep learning works are specifically designed to relight human portraits [109, 135, 69, 76]; however most works focus on relighting at a fixed viewpoint and cannot change the viewpoint. While other works can jointly do relighting and view synthesis [9, 70, 29, 134], they require complex capture setups to acquire a large number images under controlled lighting. In contrast, our approach enables realistic relighting and view synthesis for human portraits using only a sparse set of input images under natural illumination.

4.2.3 View Synthesis

View synthesis has been studied by the computer graphics and vision community for decades. Early work [16, 68] utilizes view interpolation to render novel views from neighboring source views. Other classical methods, including light fields [59, 34] and image-based rendering [21, 12, 105], have also been proposed to address view synthesis. Recently, deep learning methods [136, 73, 129, 26] have become dominant in this field of work. Learning-based methods have proven to be more expressive and they are able to represent various complex scenes with challenging visual effects. One recent work [128] combines morphable face models and generative techniques to reconstruct 3D portraits for rendering with changing viewpoints. However, it is non-trivial to extend these view synthesis techniques to support relighting at the same time. We propose a novel neural rendering approach that can simultaneously do relighting and view synthesis.

4.2.4 Neural Rendering

In addition to the aforementioned view synthesis methods, an exciting advancement is neural radiance field (NeRF) [74], which encodes a 3D scene in a compact 5D continuous

radiance field function represented by a multi-layer perceptron (MLP) and renders the radiance field using differentiable volume rendering. Concurrent works have extended NeRF to render human portraits [82, 122, 63]; however most of them rely on overfitting the network to a single target as is done in the original NeRF, which does not generalize to other unseen portraits. Other recent works [132, 120] leverage CNNs to perform per-view radiance field reasoning, leading to a generalizable neural model for view synthesis. Our approach is inspired by these CNN-based radiance field estimation techniques; we introduce novel light transport estimation modules in the per-view scene reasoning, enabling relighting in the neural volumetric rendering process.

Previous neural rendering methods have also achieved relighting. Some methods leverage controlled lighting to achieve reflectance estimation with per-scene optimization [8, 7]. Our approach instead supports relighting under natural illumination, jointly estimating lighting and lighting transport functions. Other concurrent works [107, 65] can also change the lighting conditions but still rely on per-scene optimization with a large number of images. Our approach achieves a generalizable neural network that learns specific human portrait shape and appearance priors from large training datasets. Additionally, our approach enables efficient portrait relighting and view synthesis from only a sparse set of input images.

4.3 Method

In this Chapter, we present Neural Light-transport Field (NeLF) to solve the problem of simultaneous portrait view synthesis and relighting. Our system takes as input a small set of 5 images of an unseen human face, which are captured roughly from the frontal view of the portrait. We assume the captured portraits are lit by the same distant light, which can be modeled by an environment map.

Given the captured images as well as the corresponding camera parameters, we produce a volumetric field of *light-transport* (Sec. 4.3.1). Each point in the 3D scene has a volume density σ , and a light-transport T . The light-transport is a vector whose dot-product with the

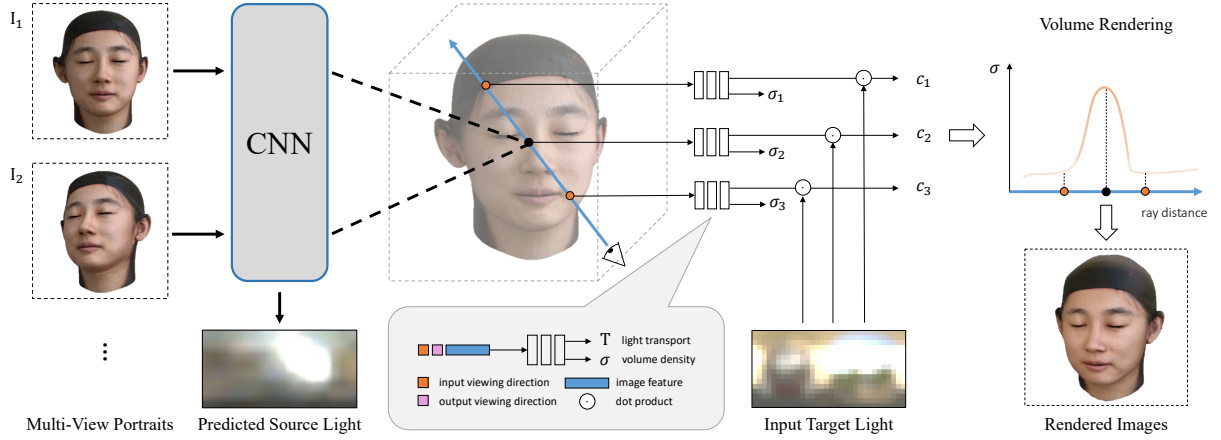


Figure 4.2. Overview of our method. The proposed algorithm takes multi-view portraits as input and predicts the source environment map, light-transport and volume density at a query point. We then use the predicted light-transport and volume density to perform the joint task of view synthesis and relighting.

global environment map produces the outgoing radiance at the 3D point. The predicted neural light-transport field (NeLF) enables us to perform view synthesis and relighting on the captured portraits. Given a new camera view and a new lighting environment, we can use the predicted light-transport to compute the outgoing radiance of each 3D point, and use a volume rendering algorithm [74] to render novel views of the captured human face under the new lighting.

We show an overview of our method in Fig. 4.2 and a detailed illustration in Fig. 4.3. In Sec. 4.3.1, we briefly review the mathematical definition of the light-transport. Section 4.3.2 describes how we predict the volume density and the light-transport at each 3D point, and Sec. 4.3.3 explains our detailed volume rendering algorithm. We show our novel domain adaptation module in Sec. 4.3.4. Finally, the implementation details are included in Sec. 4.3.5.

4.3.1 Light Transport

Light transport of a 3D point describes the relationship between its radiance and the global lighting environment. The outgoing radiance L_r of each 3D point $\mathbf{x}$ can be computed

according to the rendering equation [42]:

$$L_r(\mathbf{x}, \omega_o) = \int_{\omega_i} \rho(\mathbf{x}, \omega_i, \omega_o) (\mathbf{n} \cdot \omega_i) L_i(\mathbf{x}, \omega_i) d\omega_i, \quad (4.1)$$

where ρ is the BSDF function, $\mathbf{n}$ is the normal direction at the 3D point, and ω_i, ω_o are the incoming and outgoing ray directions, respectively. Traditional path tracing computes the radiance L_i recursively until the ray hits the global environment L . However, after unravelling the recursion, the radiance has a linear relationship with the environment map:

$$L_i(\mathbf{x}, \omega_i) \propto L(\omega'_i). \quad (4.2)$$

Here, the incoming radiance L_i of the point $\mathbf{x}$ at each incoming direction ω_i is proportional to the corresponding global environment light $L(\omega'_i)$ at each direction ω'_i . Notice that this linear relation has modeled all the paths that come from the environment to this point, modeling effects caused by shadowing, inter-reflection, material properties of the ray bounces in between, etc. If we bring Eqn. 4.2 back to Eqn. 4.1, we can simplify Eqn. 4.1 as a linear integration:

$$L_r(\mathbf{x}, \omega_o) = \int_{\omega_i} T(\mathbf{x}, \omega_o, \omega_i) L(\omega_i) d\omega_i. \quad (4.3)$$

Here, the *light transport* T has already encoded all the complexity from the environmental light L to the outgoing radiance L_r at $\mathbf{x}$.

As shown in Eqn. 4.3, given the light transport T , we can compute the outgoing radiance at each 3D point $\mathbf{x}$ under a novel environment map L using an integration, and then efficiently render the scene under arbitrary lighting environments. This type of method is called Precomputed Radiance Transfer (PRT) [106], which has been well studied in traditional computer graphics [114, 79]. In our method, we express the global lighting simply as a 8×16 environment map $\mathbf{L}$. We predict a light-transport vector $\mathbf{T}$ at each 3D point $\mathbf{x}$ given an input viewing direction. $\mathbf{T}$ also shares the same size as the environment map $\mathbf{L}$. As a result, we can simplify Eqn. 4.3 to a

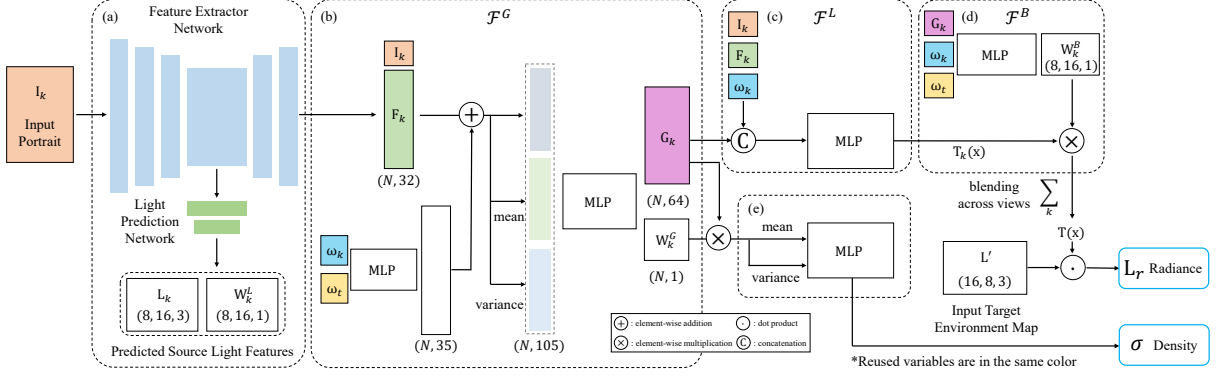


Figure 4.3. Detailed network structures of our proposed algorithm. (a) For an input image I_k , we first extract its lighting feature L_k , weights W_k^L and image feature F_k (Sec. 4.3.2). (b) Then, as described in Sec. 4.3.2, we aggregate features from different views to extract the geometry feature, G_k , and weights W_k^G . From here, our network diverges into two paths: the first part (c) predicts the per-view light transports T_k using a per-view MLP $\mathcal{F}^L$ and then (d) aggregates them with a blending MLP $\mathcal{F}^B$ to generate the full light transport T ; the second part (e) predicts the volume density σ using an MLP from the multi-view statistics of G_k and W_k^G .

dot product on each color channel:

$$L_r = T \cdot L. \quad (4.4)$$

This is similar to the light transport matrix in image-based relighting [20], which models per-pixel light transport. In our work, we compute the radiance at each 3D point following Eqn. 4.4, and later render the radiances into pixel values following the neural volumetric rendering equation in Eqn. 4.10.

4.3.2 Neural Light-transport Field

Given N different views of a human face, we use a neural network to predict a volumetric light-transport field. The key concept of our algorithm is to predict the light-transport vector T of each 3D point by aggregating the information from the input portraits. To this end, we utilize the ideas of the recent image-based rendering techniques [120] to predict the light-transport. Our system consists of two parts (see Fig. 4.3): a convolutional neural network (CNN) and several multi-layer perceptrons (MLP). The CNN operates on the captured portraits to extract the image features and predict the source lighting condition. On the other hand, the MLPs predict the

light transport $\mathbf{T}$ as well as the volume density σ for each 3D point in the scene, using multiple features. Then for any given target environment map $\mathbf{L}'$, we can simple calculate the color with Equ. 4.4 and use it for volumetric rendering (Sec. 4.3.3).

Image Feature Extraction and Source Light Prediction

We apply an U-Net style convolutional neural network [93] to N captured images. For an input portrait $\mathbf{I}_k$ at viewpoint k , the CNN extracts the image feature $\mathbf{F}_k$, as well as the source lighting feature $\mathbf{L}_k$ and confidence weights $\mathbf{W}_k^{\mathbf{L}}$ (see Fig. 4.3(a)). The size of image feature $\mathbf{F}_k$ is half of the original input image $\mathbf{I}_k$, and it is later used for light transport prediction.

Each lighting feature consists of two parts: a predicted source environment map $\mathbf{L}_k$, and a per-direction confidence map $\mathbf{W}_k^{\mathbf{L}}$ [109]. We predict the confidence map $\mathbf{W}_k^{\mathbf{L}}$ for all cameras, as each camera covers only parts of the portrait, providing partial lighting information. For example, it is less accurate to predict the light coming from the left by looking at the right side of the face. To resolve this issue, we can merge the lighting predictions $\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_N$ with their corresponding confidence maps $\mathbf{W}_1^{\mathbf{L}}, \mathbf{W}_2^{\mathbf{L}}, \dots, \mathbf{W}_N^{\mathbf{L}}$ as weights.

Nonetheless, another issue is that the network is unaware of the camera pose when predicting the lighting environment. As a result, the network is only able to predict the lighting relative to each camera, and each predicted environment-map is defined in its corresponding camera coordinate system. In order to align multiple light predictions from different coordinate systems and merge them into a global environment map, we define a rotation operator $\mathcal{R}$ that rotates them to a canonical world coordinate, provided camera extrinsics. Finally, we compute the global lighting environment with a weighted average:

$$\mathbf{L} = \frac{\sum_k \mathcal{R}(\mathbf{L}_k \odot \mathbf{W}_k^{\mathbf{L}})}{\sum_k \mathcal{R}(\mathbf{W}_k^{\mathbf{L}})}, \quad (4.5)$$

where $\odot$ means element-wise multiplication.

Volume Density and Light-transport Prediction

We use multiple MLPs to predict the volume density $\sigma(\mathbf{x})$ and the light transport $\mathbf{T}(\mathbf{x})$ at each point $\mathbf{x}$ observed from the target camera t . We first project the point $\mathbf{x}$ to all source cameras to acquire the corresponding image features $\mathbf{F}_k(\pi_k(\mathbf{x}))$, where π_k denotes the projection to camera k . We also compute source viewing direction ω_k and target viewing direction ω_t . The image features, together with ω_k and ω_t , are fed into MLPs, $\mathcal{F}^G$, to extract the multiview-aware geometry feature $\mathbf{G}_k$ and a corresponding weight $\mathbf{W}_k^G$ for each source view (see Fig. 4.3):

$$\mathbf{G}_k, \mathbf{W}_k^G = \mathcal{F}^G(\omega_t, \{\omega_k\}_{k=1}^N, \{\mathbf{F}_k(\pi_k(\mathbf{x}))\}_{k=1}^N). \quad (4.6)$$

Shown in Fig. 4.3(b), the per-view geometry feature is extracted using a PointNet style MLP structure [87], using the per-element mean and variance of the feature as additional inputs (see Fig. 4.3). This shares the idea of traditional image-based rendering [68]: the projected image features from multiple cameras should be consistent around the actual object surface. In our setup, $\mathcal{F}^G$ compares the input feature vector with its per-element mean and variance, and learns to assign more weight $\mathbf{W}_k^G$ to the best matched view. This geometry feature is used to predict the volume density $\sigma(\mathbf{x})$ of the query point using another MLP as shown in Fig. 4.3 (e).

It is highly challenging to predict the light transport vector for the novel viewing direction. Therefore, we instead predict per-view light transports using an MLP $\mathcal{F}^L$ (Fig. 4.3(c)) and then blend them for the novel view (Fig. 4.3(d)). While we can apply $\mathcal{F}^L$ to directly regress the light transport vector, this can easily lead to the network memorizing the portrait appearance in the synthetic training set. We propose to let $\mathcal{F}^L$ regress scales relative to the pixel colors; the predicted scales have the same dimensions as the light transport, and compute the light transport vector by multiplying by the pixel colors. This design effectively retains the high-frequency information in the original input images and enables better generalizability of our network to unseen real portraits (see Fig. 4.5). In particular, the light-transport vector of the source view k at

point $\mathbf{x}$ is computed by:

$$\mathbf{T}_k(\mathbf{x}) = \mathbf{I}_k(\pi_k(\mathbf{x})) \cdot \mathcal{F}^L(\omega_k, \mathbf{G}_k, \mathbf{F}_k(\pi_k(\mathbf{x}))), \quad (4.7)$$

Note that, to ensure view-consistent light transport estimations, we leverage the multiview-aware feature $\mathbf{G}_k$ in this per-view light transport prediction.

We repeat this operation to acquire $\mathbf{T}_1(\mathbf{x}), \mathbf{T}_2(\mathbf{x}), \dots, \mathbf{T}_N(\mathbf{x})$. These light-transport functions encode the information at the same 3D point, but are observed from different angles. We then calculate final target light-transport $\mathbf{T}(\mathbf{x})$ as a linear combination of $\mathbf{T}_1(\mathbf{x}), \mathbf{T}_2(\mathbf{x}), \dots, \mathbf{T}_N(\mathbf{x})$. We use another MLP $\mathcal{F}^B$ to predict the blending weights

$$\mathbf{W}_k^B = \mathcal{F}^B(\omega_k, \omega_t, \mathbf{G}_k), \quad (4.8)$$

and linearly combine the light-transports of the source views to get the final light-transport

$$\mathbf{T}(\mathbf{x}) = \sum_k \mathbf{W}_k^B \odot \mathbf{T}_k(\mathbf{x}). \quad (4.9)$$

4.3.3 Efficient Volume Rendering

For a given target environment map $\mathbf{L}'$, we follow Equ. 4.4 to compute the radiance $\mathbf{L}_r(\mathbf{x})$. The pixel value $\mathbf{I}_t$ of the novel viewpoint can be computed using the volumetric rendering equation from the original NeRF formulation [74]. Suppose the point at depth u is $\mathbf{x}(u) = \mathbf{x}_c + u \cdot \omega_c$, where $\mathbf{x}_c$ is the camera location and ω_c is the ray direction, we have:

$$\mathbf{I}_t = \int_{u_n}^{u_f} \exp\left(-\int_0^u \sigma(\mathbf{x}(v)) dv\right) \sigma(\mathbf{x}(u)) \mathbf{L}_r(\mathbf{x}(u)) du. \quad (4.10)$$

To be more specific, we integrate along the light ray with u_n, u_f as the near and far bounds, respectively. We additionally predict the depth and the alpha channel, which are then used for

supervision (Sec. 4.3.5):

$$\begin{aligned}\mathbf{D}_t &= \int_{u_n}^{u_f} \exp\left(-\int_0^u \sigma(\mathbf{x}(v))dv\right) \sigma(\mathbf{x}(u))u \cdot d\mathbf{u}, \\ \mathbf{A}_t &= \int_{u_n}^{u_f} \exp\left(-\int_0^u \sigma(\mathbf{x}(v))dv\right) \sigma(\mathbf{x}(u))d\mathbf{u}.\end{aligned}\tag{4.11}$$

The original NeRF paper computes the radiance at each sample point. However, most of the sample points are 0 when the scene is spatially sparse. This creates a lot of redundancy when rendering a new image. In our setup, since portraits are usually sparse, we exploit the concept of *visual hull* to prune out the queries that are unnecessary. From the input portraits, we utilize a portrait matting algorithm [48] to extract their mask $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_N$. Then for each 3D point $\mathbf{x}$ viewed by the target camera, we project its position onto all the masks, and query the MLPs if all the projections onto $\mathbf{M}_k(\pi_k(\mathbf{x}))$ are nonzero. Otherwise, we directly set the light transport $\mathbf{T}(\mathbf{x})$ and the density $\sigma(\mathbf{x})$ to be 0. This is essentially restricting the network to learn within the visual hull defined by the silhouette of the portraits. By doing this, we achieve faster convergence during training and more efficient rendering during inference.

4.3.4 Domain Adaptation

We train our network on synthetically rendered human face data (see Sec. 4.4.1). This data does not model the distribution of the real human faces very well. Thus, naively training our network using the rendered faces results in poor generalizability on real portraits. It is optimal to include real multi-view portraits in our training data. However, there is no publicly available large-scale human face dataset and it is also challenging to capture real portraits under different viewpoints and lighting without a light stage.

To this end, we propose a novel domain adaptation module that effectively enhances the generalizability of our network by augmenting the CNN feature extractor with a large number of real portrait images in the CelebAMask-HQ dataset [58]. These real images do not have groundtruth labels of their lighting and camera parameters; however we show that they can be

effectively used to regularize our feature extractor to adapt to the distribution of real portrait images. In particular, for each training iteration, we additionally feed in a real portrait into the feature extractor and get an image feature. Rather than using the image feature to predict the light transport and perform volumetric rendering, we append 2 more convolution layers to the U-Net to directly recreate the original image, enhancing the expressiveness of the feature extractor. With this joint training strategy, we can equip the feature extractor with stronger capabilities to reason about real portrait images and avoid overfitting to the biased distribution of the synthetic portrait dataset.

4.3.5 Implementation Detail

In general, there are two kinds of relighting tasks we need to solve: changing to a completely novel environment light, or rotating the original light. In order to solve both tasks in NeLF, we implement two modes during training: novel light mode and self rotation mode. In the novel light mode, we provide a new lighting as the target light to the MLP to compute the radiance. In the self rotation mode, we predict the lighting of the input portrait from the CNN, rotate the predicted light for a certain angle, and use that as the target light. We allocate 70% of the training for novel light, and the rest for self rotation.

Our network is supervised with multiple losses. We implement a rendering loss $\mathcal{L}_c$, a depth loss $\mathcal{L}_d$, an alpha mask loss $\mathcal{L}_a$, a lighting loss $\mathcal{L}_l$ and an image consistency loss $\mathcal{L}_p$. To begin with, the rendering loss $\mathcal{L}_c$ is simply an $\mathcal{L}_1$ loss on the predicted RGB $\mathbf{I}_t$ such that it is as close as possible to the ground truth value $\mathbf{I}_{gt}$

$$\mathcal{L}_c = ||\mathbf{I}_t - \mathbf{I}_{gt}||_1. \quad (4.12)$$

We additionally supervise the predicted depth and the alpha channel using the ground truth values. This is to make sure the predicted shape is meaningful and obeys multi-view constraints.

These losses are formulated as:

$$\mathcal{L}_a = \|\mathbf{A}_t - \mathbf{A}_{gt}\|_1, \mathcal{L}_d = \frac{1}{d} \|\mathbf{D}_t - \mathbf{D}_{gt}\|_1, \quad (4.13)$$

where $d = 200mm$ is the average size of a human head. For lighting, we use log- $\mathcal{L}_1$ loss on the predicted environment map L and the ground truth environment map $\mathbf{L}_{gt}$:

$$\mathcal{L}_l = \|\log(1 + \mathbf{L}) - \log(1 + \mathbf{L}_{gt})\|_1 \quad (4.14)$$

As mentioned in Sec. 4.3.4, we train our network jointly with a self consistency constraint on the image encoder to ensure that the CNN can learn meaningful representations of real human faces. For each iteration, we randomly pick a in-the-wild portrait $\mathbf{I}_c$ from the CelebAMask-HQ dataset. We feed the image only into the feature extractor to get a reconstruction of the same image $\hat{\mathbf{I}}_c$. We supervise the reconstruction using an image consistency loss $\mathcal{L}_p$:

$$\mathcal{L}_p = \|\hat{\mathbf{I}}_c - \mathbf{I}_c\|_1. \quad (4.15)$$

Notice that we only apply the image consistency loss on the random in-the-wild portrait $\mathbf{I}_c$. We don't enforce the image encoder to reproduce the calibrated multi-view images. Together, the final loss $\mathcal{L}_{total}$ is a sum of all above-mentioned losses:

$$\mathcal{L}_{total} = \mathcal{L}_c + \mathcal{L}_d + \mathcal{L}_a + \mathcal{L}_l + \mathcal{L}_p. \quad (4.16)$$

The detailed network structures of our CNN and MLP are shown in Fig. 4.3. We implement our method in PyTorch [83]. We use Adam [52] as our optimizer and the learning rates are set to 10^{-4} for the MLPs and 2×10^{-4} for the image encoder. Our network is trained on 4 NVIDIA 2080Ti GPUs for 300k steps, which takes around 36 hours.

Table 4.1. Quantitative comparison on synthetic evaluation dataset. Our method performs better in both PSNR and SSIM over baseline methods, providing better visual quality. For comparison against direct prediction of light transport, while the visual artifacts are disturbing in Fig. 4, they reflect only a small part of the image, and therefore do not have as much of an impact on the PSNR and SSIM scores.

Method	PSNR SSIM
SIPR + IBRNet	33.55 0.8826
IBRNet + SIPR	33.18 0.8611
Ours w/o encoder augmentation	33.54 0.8916
Ours w/ direct light transport	33.70 0.8928
Ours	33.61 0.8922

4.4 Results

4.4.1 Data

In order to train our novel representation, we choose the FaceScape dataset [131] as it provides a good collection of realistic human head models with high-resolution texture and displacement maps. Due to privacy issues, some of the models have blurred textures around the eyes. Thus, we select 360 unblurred models from the whole dataset and use Blender [11] with the Cycles renderer to generate the training and testing data images in 512×512 resolution. To be more specific, we generate data triplets that contain (a) 5 source views with the first one being the frontal view, (b) a novel view with the same lighting but randomly rotated, and (c) a novel view with lighting randomly selected from a pool of environment maps. (a) is used as input to our network, (b) is the self rotation mode supervision, and (c) is the novel light mode supervision. Both (b) and (c) also provide novel view supervision. The novel views are uniformly sampled within 30° in azimuthal and 30° in elevational angle from the frontal face view. The distance of the cameras is chosen randomly from 100cm to 200cm to simulate real life captures. We also adjust the field-of-view accordingly so that the face would take up most space. This training triplet design allows for learning of both geometric and lighting information.

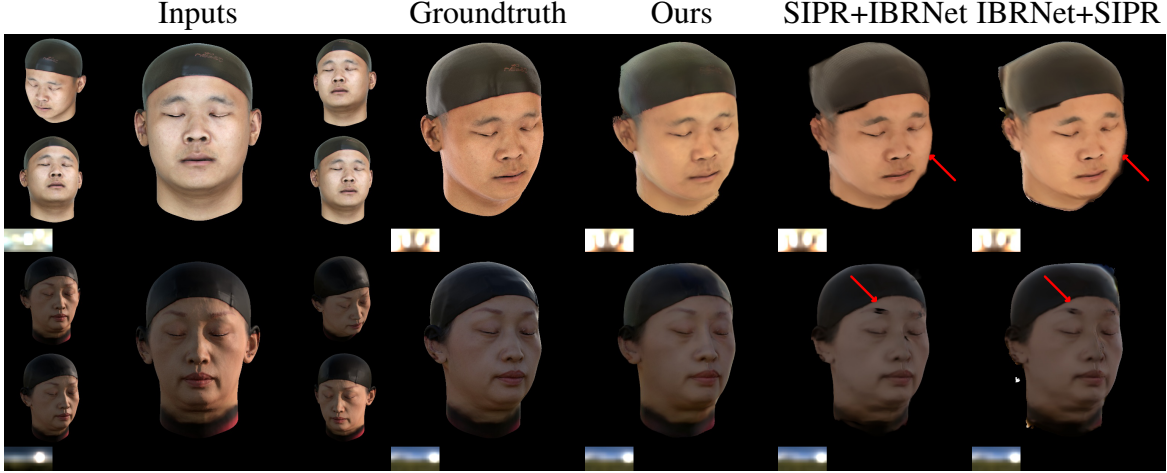


Figure 4.4. We compare our results of simultaneous view synthesis and relighting with possible baseline methods on unseen samples from our synthetic evaluation dataset. Baseline methods fail to provide accurate geometry and contain random hole artifacts. For instance, SIPR + IBRNet shows some hole artifacts near the nose area and IBRNet + SIPR shows blurry results around the cheek in the top right image. Please refer to the supplementary video for more results.

4.4.2 Comparison with previous methods

To the authors’ best knowledge, our algorithm is the first to achieve simultaneous view synthesis and relighting on unseen subjects from a sparse set of multi-view images under natural illumination. SIPR [109] can perform relighting on unseen portraits given a single image, while IBRNet [120] achieves view synthesis on novel scenes. We combine these previous two state-of-the-art algorithms and use the combinations as the baseline of our algorithm. There are two possible ways to combine: first do the relighting on the input views, and then synthesize the new view (SIPR+IBRNet); or first do view synthesis and then change the lighting of the synthesized image (IBRNet+SIPR). Notice that we have to correctly rotate the target light to each view in order to align the target light with the camera.

We retrain the network of IBRNet and SIPR on our synthetic dataset for fair comparisons. We choose 4 subjects out of the total 360 identities to serve as our evaluation dataset. Table 4.1 shows the quantitative comparison of these two baselines, as well as our method. Our method outperforms both of the baselines with both higher PSNR and SSIM. In addition, we observe

Table 4.2. Quantitative comparison on the task of relighting and view-synthesis individually on synthetic dataset. Our method performs comparably, but slightly worse than the state-of-the-art on each task.

	Ours	SIPR	IBRNet
Relighting	0.9053	0.9279	—
View-synthesis	0.8683	—	0.8949

that the baseline methods often lead to obvious visual artifacts in their renderings as shown in Fig. 4.4 (please see our supplementary video for more examples). Our approach instead can achieve much higher visual quality, consistently producing realistic renderings across different lighting and viewpoints.

These two baselines fail for different reasons. If relighting the input images first, the relighting algorithm might perform slightly differently on each input view, which will break the multi-view consistency for view synthesis. On the other hand, doing the relighting after the view synthesis also does not work well, since the relighting algorithm now has no access to the multi-view information. In contrast, our approach learns to effectively aggregate multi-view appearance features to predict the light transport of each shading point, leading to accurate relighting effects that are consistent across multi-view viewpoints. Our approach can simultaneously achieve high-quality relighting and view synthesis.

We have also evaluated our method on relighting and view-synthesis respectively using our synthetic dataset. We perform relighting by rendering the view at the frontal camera under new lighting, and do view synthesis by first estimating the original light, and then relighting under the predicted lighting from new views. As shown by the SSIM values in Tab. 4.2, our method performs comparably, but slightly worse than the baselines on each task. This is because our main goal is not to improve individual performance, but to enable the combination of these two tasks. Thus, our view-synthesis performance also includes the error from lighting estimation and relighting. Even though SIPR and IBRNet perform slightly better in their individual tasks, their combination does not work as well as ours as shown in Fig. 4.4. Our method aggregates multi-view features to predict the light-transport on each shading point, leading to multi-view

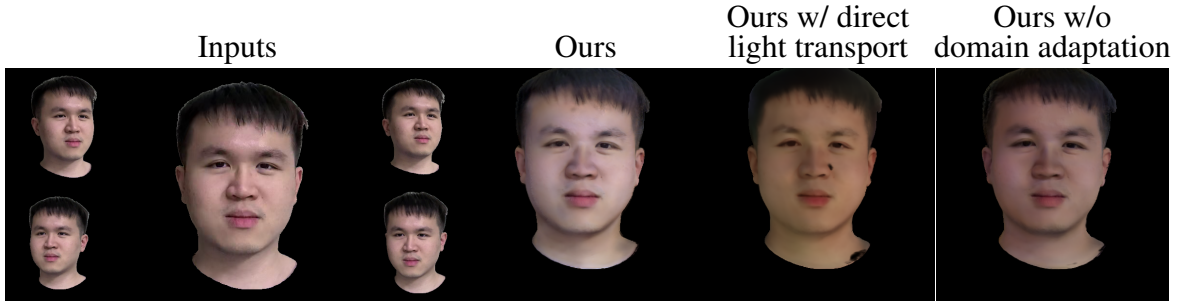


Figure 4.5. We compare our method with two possible ablations on a real captured image. We show that by modulating the light transport prediction and applying our domain adaption method, our method can generalize well to real portraits with minimum color shift or artifacts.

consistent relighting effects.

4.4.3 Ablations

We ablate our algorithm by directly predicting the light transport rather than modulating the prediction using the pixel value in Eqn. 4.7. The quantitative results of synthetic validation set and qualitative results of real portraits are shown in Tab. 4.1 and Fig. 4.5 respectively. Note that, although the network with direct light transport achieves slightly higher performance on the validation dataset, our full model performs much better on the real portraits as shown in Fig. 4.5. In particular, directly predicting the light-transport can lead to artifacts, for instance the black hole around the nose area in the portrait. In addition, the method introduces more color shift compared to our full algorithm. Our full model learns to predicts scales of the original pixel colors; this retains the original color signals and can generalize better to unseen real portraits.

In addition, we show the effect of our domain adaptation module in Fig. 4.5. We show that with the additional adaptation module, our network is able to generalize to unseen color distributions much better than without this module. As can be seen from the figure, without the adaptation, the network fails to recreate the vibrancy of the input images. Moreover, the left part of the portrait exhibits more orange color than our proposed method. Our domain adaptation module essentially regularizes the network to be able to better reproduce the appearance in the original input images. This is not only helpful when testing on real portrait images, but can also

Table 4.3. Quantitative comparison on different input view counts. We show that 5 views provide the best possible results in both PSNR and SSIM metrics. Visual quality starts to degrade as view count reduces.

View Num	PSNR	SSIM
2 Views	30.91	0.6388
3 Views	32.77	0.8177
4 Views	33.40	0.8731
5 Views	33.61	0.8922

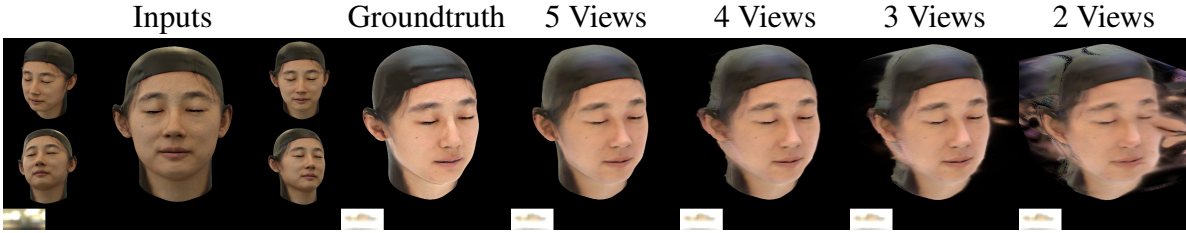


Figure 4.6. Qualitative evaluation on the effects of different input view counts. We show that there is a small difference between 4 and 5 views, while 2 and 3 views result in more artifacts due to occlusions and the lack of stereo coverage.

improve our performance on the synthetic validation set as shown in Tab. 4.1.

We also study the effects of input view numbers on the rendering quality. Our network is originally trained with 5 images as the input views. We test our network on unseen subjects from the validation dataset by feeding 2, 3, and 4 views into the network for view synthesis and relighting. Table 4.3 shows the quantitative evaluation and Figure 4.6 shows the results. Our network fails to render meaningful content when only 2 images are given. However, as we have 3 or more images as the input views, our network can render new views under new lighting with high quality.

4.4.4 Real Portraits

In order to show the performance of our method on real data, we capture real portraits with smartphones by shooting a video clip of the subject. We employ COLMAP [97, 96] to recover camera poses and rough depth statistics to determine near and far planes. We also normalize the camera poses such that the origin is approximately the head center. This is done

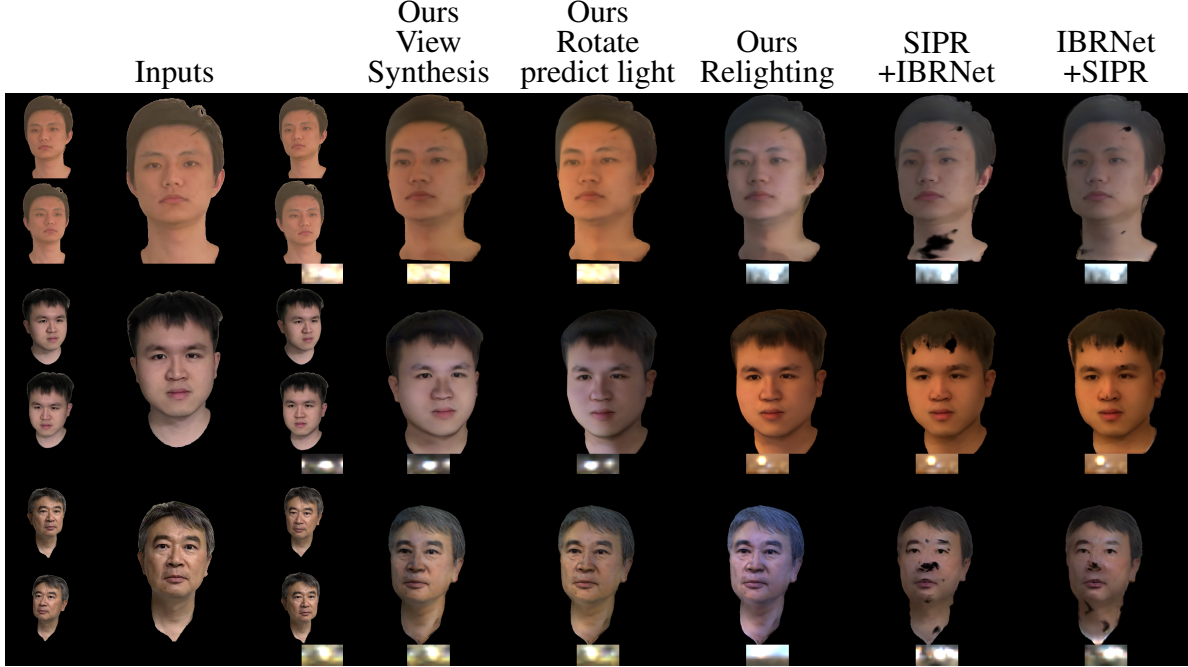


Figure 4.7. Qualitative results on real portraits. Our method produces better visual quality than the baseline methods of SIPR and IBRNet. For example, in the forehead and neck regions, baseline methods produce hole artifacts, whereas our proposed method is able to synthesize realistic novel images under new lighting conditions.

by first extracting the center of facial landmarks with Bulat et al. [13], and estimating the face direction to calculate the head center. We apply MODNet [49] to clean up the background and use BiSeNet [133] to further remove garments and keep the face portion. As our training data does not contain any clothes, this can ensure that the rendered results are not perturbed by unrelated information.

The results are shown in Fig. 4.1 and 4.7. We can observe that our method achieves state-of-the-art quality on the joint task of view synthesis and relighting. Our proposed method is able to infer source environment map and render novel view portraits with the rotated source light. Additionally, we can input a target environment map to control the new lighting condition. For the task of rendering real portraits, the comparison baselines (both SIPR+IBRNet and IBRNet+SIPR) lead to obvious artifacts with many visible holes, more obvious than their artifacts on the synthetic validation set; this is because the baseline methods’ networks overfit to the training set’s data

distribution and the same issue of inconsistency between separate relighting and view synthesis modules (as discussed in Sec. 4.4.2) becomes more significant on the real data. Thanks to our effective modeling of the light transport in the 3D space and our domain adaptation module, our model can synthesize realistic relighting and view synthesis results on the real portrait images, leading to significantly better results than the baseline methods.

4.4.5 Limitations

Although our proposed method generates photo-realistic rendering results, it still possesses some limitations. For example, since we train on a synthetic dataset, our model might exhibit some color shifts in certain cases when tested on real portraits. This can be ameliorated by training on a real multi-view portrait dataset. Another issue is the slight blurriness in our rendered results. This is possibly caused by the limited network capacity, which can be further increased to allow extraction of image features with higher resolution. In addition, we downsample the feature map to allow for faster training and inference. With more computational resources, it is possible to use full-scale image features. Last, because our training dataset have a global specular coefficient instead of dedicated specular maps, some complex view-dependent effects such as glints are not well reconstructed in the training images. As a result, our method fails in some cases where high-frequency specular highlights are presented.

4.5 Conclusions and Future Work

In this chapter, we tackle the joint problem of portrait view synthesis and relighting, which prior works fails to handle. We solve this problem by introducing the *neural light-transport field*, which encodes the volume density and light transport vectors of each 3D point in the scene, enabling relighting with a target environment map. We demonstrate that with only 5 input views, our method is able to generalize across unseen portraits and produce better portrait renderings than previous approaches built specifically for either view synthesis or relighting.

In future work, we would like to explore the possibility of even fewer input images.

We show that our method degrades for a smaller number of views. This is caused by the depth ambiguities and unseen areas, which requires prior knowledge to hallucinate reasonable renderings. Another possible direction is animated relightable avatars as our proposed method does not handle a talking head explicitly. All in all, we believe that the joint problem of view synthesis and relighting is crucial in immersive applications like mixed reality, and our work can advance the field in that direction.

This chapter is a reformatted version of the material as it appears in “NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting”, Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, and Ravi Ramamoorthi, Eurographics Symposium on Rendering, 2021 [110]. The dissertation author was the co-primary investigator and author of this paper.

Chapter 5

Conclusion

In this dissertation, we develop efficient and practical portrait relighting algorithms that enable relighting on casually captured images. In Chapter 2, we propose single image portrait relighting. We use a light stage to collect a small dataset of relightable portraits, and train a neural network end-to-end with the generated portrait pairs under different natural lightings. Our network enables changing the lighting condition of a single portrait taken from the cellphone within real-time response. We achieve convincing relighting results for two reasons: we train our network with real captured portraits and we train the network end-to-end without any assumption on appearance models.

Single image portrait relighting can change the lighting into natural illuminations, but fails on high-frequency lightings such as point lights. To solve this problem, in Chapter 3, we investigate how to super-sample the number of lights on the light stage, in order to achieve continuous high-frequency relighting. Since naively training a network will cause flickering and ghosting shadows, we introduce dropout-like active set selection and develop alias-free pooling techniques. Using our light stage super-resolution technique, we can relight the portrait accurately under high-frequency lighting conditions. We additionally study when this approach is necessary given the frequency of the lighting. Although this approach still requires the portrait to be captured under the light stage, it enables higher quality relighting of human portraits as the groundtruth, which benefits the downstream applications that rely on the light stage.

In Chapter 4, we push forward our relighting techniques toward view synthesis, and propose neural light-transport field (NeLF) for simultaneous view synthesis and relighting. Given 5 portraits casually captured from the frontal view under natural illumination, we develop a network that encodes the portrait into a volumetric field of light-transport vectors, which later enables re-rendering from arbitrary view under any natural lighting. Our network is trained on a synthetic portrait dataset, but can generalize well on portraits of real people.

We have proposed several methods that push forward the efficiency and the practicality of human portrait relighting. Now I would like to discuss some of the promising future directions generally on portrait appearances.

Appearance representation

There still does not exist a consensus on a representation for portrait appearances that is accurate, flexible, and also physically meaningful. Recent advances on model-based face inverse rendering have enabled us to recover a full face model from a single image. However, these models only contain the face part. More accurate models of eyes, teeths, human hairs are needed to fully reconstruct the human faces. More recently, NeRF [74] has drawn a lot of attention in academia. It produces very accurate results, while also being physically interpretable and compact. We believe NeRF can be a very good representation for portrait appearances, as it contains the advantages of both model-based and abstraction-based methods. It follows the traditional volumetric rendering pipeline, while abstracting the scene representation using a very simple neural network. However, how to efficiently model a portrait using volumetric representations still remains an unsolved problem. The method NeLF we proposed in Chapter 4 is the first step in this direction, but how to resolve the ambiguity and how to render the volume efficiently still remain open problems.

Efficient acquisition

The light stage has been used in most of our portrait relighting project. Thus another possible direction to explore is how we can understand the portrait appearances with less-

expensive captures. The light stage is of course a very useful tool for appearance acquisition, but capturing people inside a light stage requires a huge amount of effort to do the setup, not to mention the cost to build and maintain a light stage. We hope we can find a better way to use casually captured data to better understand human face appearances.

Portrait prior

In the portrait relighting projects we presented in this thesis, we only utilized a weak prior of human portraits, which was implicitly learned by the neural networks. However, we believe we haven't fully exploited all the visual cues from the portrait. Thus, a promising direction is to find more implicit or explicit priors of human faces, and find ways to combine them with the neural network.

Dynamic portraits

In the thesis, we are solving the problem of relighting on static portraits. However, people are dynamic, and constantly making movement and changing facial expression, which gives much more information than just a single image. It is also a very promising direction to explore how to aggregate information of the portrait from consecutive frames with different facial expressions.

Bibliography

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. *OSDI*, 2016.
- [2] Apple. Use portrait mode on your iphone. <https://support.apple.com/en-us/HT208118>, 2017.
- [3] Jonathan T. Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. Fast bilateral-space stereo for synthetic defocus. *CVPR*, 2015.
- [4] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2014.
- [5] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015.
- [6] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. The bas-relief ambiguity. *IJCV*, 1999.
- [7] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020.
- [8] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. *arXiv preprint arXiv:2007.09892*, 2020.
- [9] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5960–5969, 2020.
- [10] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. *SIGGRAPH*, 1999.

- [11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2020.
- [12] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432. ACM, 2001.
- [13] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.
- [14] Dan Andrei Calian, Jean-François Lalonde, Paulo F. U. Gotardo, Tomas Simon, Iain A. Matthews, and Kenny Mitchell. From Faces to Outdoor Light Probes. *Computer Graphics Forum*, 2018.
- [15] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *SIGGRAPH*, 2000.
- [16] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288, 1993.
- [17] Zhen Cheng, Zhiwei Xiong, Chang Chen, and Dong Liu. Light field super-resolution: A benchmark. In *CVPR Workshops*, 2019.
- [18] Paul Debevec. The light stages and their applications to photoreal digital actors. In *SIGGRAPH Asia*, 2012.
- [19] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. *SIGGRAPH*, 2000.
- [20] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’00, page 145–156, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [21] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20. ACM, 1996.
- [22] Craig Donner, Tim Weyrich, Eugene d’Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz. A layered, heterogeneous reflectance model for acquiring and rendering human skin. *SIGGRAPH*, 2008.
- [23] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. *NIPS*, 2000.

- [24] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. A frequency analysis of light transport. In *SIGGRAPH*, 2005.
- [25] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *SIGGRAPH*, 2008.
- [26] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world’s imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5515–5524, 2016.
- [27] Martin Fuchs, Hendrik PA Lensch, Volker Blanz, and Hans-Peter Seidel. Superresolution reflectance fields: Synthesizing images for intermediate light directions. In *Computer Graphics Forum*, 2007.
- [28] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. <https://arxiv.org/abs/2012.03065>, 2020.
- [29] Duan Gao, Guojun Chen, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deferred neural lighting: free-viewpoint relighting from unstructured photographs. *ACM Transactions on Graphics (TOG)*, 2020.
- [30] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gamberetto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *SIGGRAPH Asia*, 2017.
- [31] Pablo Garrido, Levi Valgaerts, Ole Rehmsen, Thorsten Thormahlen, Patrick Perez, and Christian Theobalt. Automatic face reenactment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4217–4224, 2014.
- [32] Michaël Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *SIGGRAPH*, 2017.
- [33] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.
- [34] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996.
- [35] Kaiwen Guo, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Peter Lincoln, Paul Debevec, Shahram Izad, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, and Rohit Pandey. The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting. In *SIGGRAPH Asia*, 2019.

- [36] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *SIGGRAPH Asia*, 2016.
- [37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CVPR*, 2015.
- [38] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *IJCV*, 2007.
- [39] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. *CVPR*, 2017.
- [40] Berthold K. P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, 1970.
- [41] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. *CVPR*, 2017.
- [42] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986.
- [43] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *SIGGRAPH*, 2016.
- [44] Yoshihiro Kanamori and Yuki Endo. Relighting humans: occlusion-aware inverse rendering for fullbody human images. *SIGGRAPH Asia*, 2018.
- [45] Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. Efficient reflectance capture using an autoencoder. *SIGGRAPH*, 2018.
- [46] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.
- [47] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.
- [48] Zhanghan Ke, Kaican Li, Yurou Zhou, Qiuhua Wu, Xiangyu Mao, Qiong Yan, and Rynson WH Lau. Is a green screen really necessary for real-time human matting? *arXiv preprint arXiv:2011.11961*, 2020.
- [49] Zhanghan Ke, Kaican Li, Yurou Zhou, Qiuhua Wu, Xiangyu Mao, Qiong Yan, and Rynson W.H. Lau. Is a green screen really necessary for real-time portrait matting? *ArXiv*, abs/2011.11961, 2020.
- [50] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE transactions on pattern analysis and machine intelligence*, 33(2):394–405, 2010.

- [51] Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. E-LPIPS: robust perceptual image similarity via random transformation ensembles. *CoRR*, abs/1906.03973, 2019.
- [52] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.
- [53] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- [54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.
- [55] Jean-François Lalonde, Alexei A. Efros, and Srinivasa G. Narasimhan. Estimating natural illumination from a single outdoor image. *ICCV*, 2009.
- [56] Edwin H. Land and John J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 1971.
- [57] Yann Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989.
- [58] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [59] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.
- [60] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *SIGGRAPH Asia*, 2018.
- [61] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.
- [62] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *SIGGRAPH*, 2018.
- [63] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *arXiv preprint arXiv:2103.01954*, 2021.
- [64] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

- [65] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.
- [66] Vincent Masselus, Pieter Peers, Philip Dutré, and Yves D Willemsy. Smooth reconstruction and compact representation of reflectance functions for image-based relighting. In *Proceedings of the fifteenth eurographics conference on rendering techniques*, 2004.
- [67] Wojciech Matusik, Matthew Loper, and Hanspeter Pfister. Progressively-refined reflectance functions from natural illumination. In *Rendering Techniques*, pages 299–308, 2004.
- [68] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 39–46, 1995.
- [69] Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhoefer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, Peter Denny, Sofien Bouaziz, Peter Lincoln, Matt Whalen, Geoff Harvey, Jonathan Taylor, Shahram Izadi, Andrea Tagliasacchi, Paul Debevec, Christian Theobalt, Julien Valentin, and Christoph Rhemann. Deep reflectance fields: High-quality facial reflectance field inference from color gradient illumination. *SIGGRAPH*, 2019.
- [70] Abhimitra Meka, Rohit Pandey, Christian Haene, Sergio Orts-Escolano, Peter Barnum, Philip Davidson, Daniel Erickson, Yinda Zhang, Jonathan Taylor, Sofien Bouaziz, Chloe Legendre, Wan-Chun Ma, Ryan Overbeck, Thabo Beeler, Paul Debevec, Shahram Izadi, Christian Theobalt, Christoph Rhemann, and Sean Fanello. Deep relightable textures: volumetric performance capture with neural rendering. *ACM Transactions on Graphics (TOG)*, 39(6):1–21, 2020.
- [71] Peyman Milanfar. *Super-resolution imaging*. CRC Press, 2010.
- [72] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. In *SIGGRAPH*, 2019.
- [73] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.
- [74] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.
- [75] Koki Nagano, Graham Fyffe, Oleg Alexander, Jernej Barbič, Hao Li, Abhijeet Ghosh, and Paul Debevec. Skin microstructure deformation with displacement map convolution. *SIGGRAPH*, 2015.

- [76] Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, and Andreas Lehrmann. Learning physics-guided face relighting under directional light. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5124–5133, 2020.
- [77] Thomas Nestmeyer, Iain Matthews, Jean-François Lalonde, and Andreas M Lehrmann. Structural decompositions for end-to-end relighting. *arXiv preprint arXiv:1906.03355*, 2019.
- [78] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-Frequency Shadows using Non-Linear Wavelet Lighting Approximation. In *SIGGRAPH*, 2003.
- [79] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.*, 22(3):376–381, July 2003.
- [80] Matthew O’Toole and Kiriakos N. Kutulakos. Optical computing for fast light transport. In *SIGGRAPH*, 2010.
- [81] Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *SIGGRAPH*, 2011.
- [82] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan Goldman, Steven Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. <https://arxiv.org/abs/2011.12948>, 2020.
- [83] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [84] Pieter Peers, Dhruv K Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. Compressive light transport sensing. *ACM Transactions on Graphics (TOG)*, 2009.
- [85] Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. Post-production facial performance relighting using reflectance transfer. *SIGGRAPH*, 2007.
- [86] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *SIGGRAPH*, 2004.
- [87] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

- [88] Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. Neural btf compression and interpolation. In *Computer Graphics Forum*, 2019.
- [89] Ravi Ramamoorthi and Pat Hanrahan. A Signal-Processing Framework for Inverse Rendering. In *SIGGRAPH*, 2001.
- [90] Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM TOG*, 2015.
- [91] Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM Transactions on Graphics (ToG)*, 34(4):1–12, 2015.
- [92] Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. Global illumination with radiance regression functions. *SIGGRAPH*, 2013.
- [93] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, 2015.
- [94] I Sato, T Okabe, Y Sato, and K Ikeuchi. Appearance Sampling for Obtaining a set of basis images for variable illumination. In *ICCV*, 2003.
- [95] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 2009.
- [96] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [97] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- [98] J.B. Schriever. *Complete Self-instructing Library of Practical Photography*. American school of art and photography, 1909.
- [99] Sebastian Schütze. *Caravaggio: The Complete Works*. Taschen, 2015.
- [100] Pradeep Sen and Soheil Darabi. Compressive dual photography. *Computer Graphics Forum*, 2009.
- [101] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. Sfsnet: Learning shape, reflectance and illuminance of faces in the wild. *CVPR*, 2018.
- [102] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of faces in the wild’. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [103] YiChang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, and Frédo Durand. Style transfer for headshot portraits. *SIGGRAPH*, 2014.

- [104] Zhixin Shu, Sunil Hadap, Eli Shechtman, Kalyan Sunkavalli, Sylvain Paris, and Dimitris Samaras. Portrait lighting transfer using a mass transport approach. *SIGGRAPH*, 2018.
- [105] Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. *ICCV*, 2009.
- [106] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, 2002.
- [107] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021.
- [108] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.
- [109] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on graphics (TOG)*, 38(4):79–1, 2019.
- [110] Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, and Ravi Ramamoorthi. Nelf:. In *Eurographics Symposium on Rendering 2021 (EGSR 2021)*. EGSR, 2021.
- [111] Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T Barron, and Ravi Ramamoorthi. Light stage super-resolution: continuous high-frequency relighting. *ACM Transactions on Graphics (TOG)*, 2020.
- [112] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. State of the art on neural rendering. In *Computer Graphics Forum*, 2020.
- [113] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355, 2018.
- [114] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on graphics (TOG)*, 25(3):967–976, 2006.
- [115] Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. Acquiring reflectance and shape from continuous spherical harmonic illumination. *SIGGRAPH*, 2013.

- [116] Borom Tunwattanapong, Abhijeet Ghosh, and Paul Debevec. Practical image-based relighting and editing with spherical-harmonics and local lights. In *2011 Conference for Visual Media Production*. IEEE, 2011.
- [117] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.
- [118] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *SIGGRAPH*, 2018.
- [119] J Wang, Y Dong, X Tong, Z Lin, and B Guo. Kernel Nystrom method for light transport. *ACM Transactions on Graphics*, 2009.
- [120] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.
- [121] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.
- [122] Ziyang Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhöfer. Learning compositional radiance fields of dynamic human heads. *arXiv preprint arXiv:2012.09955*, 2020.
- [123] Henrique Weber, Donald Prévost, and Jean-François Lalonde. Learning to estimate indoor lighting from 3d objects. *3DV*, 2018.
- [124] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011.
- [125] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *SIGGRAPH*, 2005.
- [126] Robert J. Woodham. Photometric Method For Determining Surface Orientation From Multiple Images. *Optical Engineering*, 1980.
- [127] Yuxin Wu and Kaiming He. Group normalization. *ECCV*, 2018.
- [128] Sicheng Xu, Jiaolong Yang, Dong Chen, Fang Wen, Yu Deng, Yunde Jia, and Xin Tong. Deep 3d portrait from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7710–7720, 2020.
- [129] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

- [130] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *SIGGRAPH*, 2018.
- [131] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [132] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020.
- [133] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.
- [134] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, Jonathan T. Barron, Ravi Ramamoorthi, and William T. Freeman. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG)*, 40(1):1–17, 2021.
- [135] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7194–7202, 2019.
- [136] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.
- [137] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer Graphics Forum*, 2018.