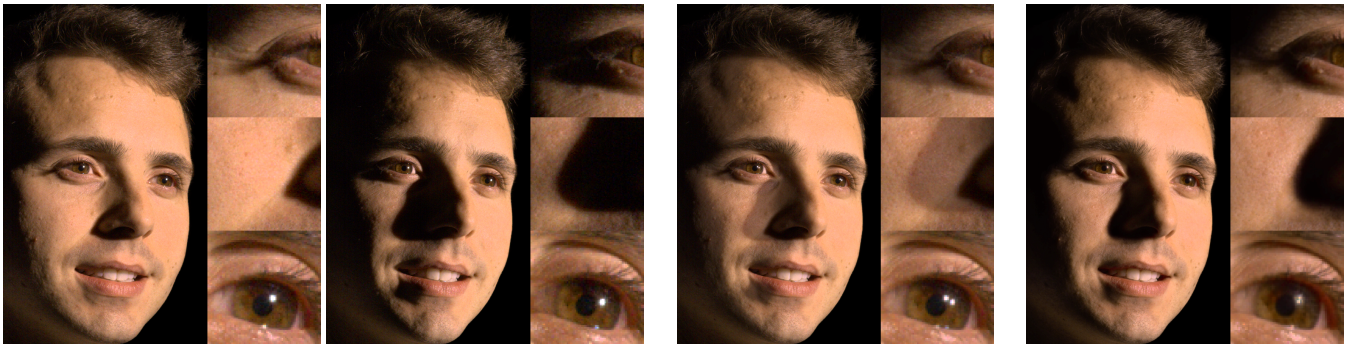
Light Stage Super-Resolution: Continuous High-Frequency Relighting

TIANCHENG SUN and ZEXIANG XU, University of California, San Diego
XIUMING ZHANG, Massachusetts Institute of Technology
SEAN FANELLO, CHRISTOPH RHEMANN, and PAUL DEBEVEC, Google
YUN-TA TSAI and JONATHAN T. BARRON, Google Research
RAVI RAMAMOORTHI, University of California, San Diego



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

Fig. 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.

The light stage has been widely used in computer graphics for the past two decades, primarily to enable the relighting of human faces. By capturing the appearance of the human subject under different light sources, one obtains the light transport matrix of that subject, which enables image-based relighting in novel environments. However, due to the finite number of lights in the stage, the light transport matrix only represents a sparse sampling on the entire sphere. As a consequence, relighting the subject with a point light or a directional source that does not coincide exactly with one of the lights in the stage requires interpolation and resampling the images corresponding to nearby lights, and this leads to ghosting shadows, aliased specularities, and other artifacts. To ameliorate these artifacts and produce better results under arbitrary high-frequency lighting, this paper proposes 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.

CCS Concepts: • **Computing methodologies** → **Image-based rendering**; **Neural networks**.

Additional Key Words and Phrases: Portrait relighting, Image-based relighting.

ACM Reference Format:
Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T. Barron, and Ravi Ramamoorthi. 2020. Light Stage Super-Resolution: Continuous High-Frequency Relighting. *ACM Trans. Graph.* 39, 6, Article 260 (December 2020), 12 pages. https://doi.org/10.1145/3414685.3417821

Authors' addresses: Tiancheng Sun, tis037@cs.ucsd.edu; Zexiang Xu, zexiangxu@cs.ucsd.edu, University of California, San Diego; Xiuming Zhang, xiuming@csail.mit.edu, Massachusetts Institute of Technology; Sean Fanello, seanfa@google.com; Christoph Rhemann, crhemann@google.com; Paul Debevec, debevec@google.com, Google; Yun-Ta Tsai, yuntatsai@google.com; Jonathan T. Barron, barron@google.com, Google Research; Ravi Ramamoorthi, ravir@cs.ucsd.edu, University of California, San Diego.

0730-0301/2020/12-ART260 $15.00
https://doi.org/10.1145/3414685.3417821

1 INTRODUCTION

A central problem in computer graphics and computer vision is that of acquiring some observations of an object, and then producing photorealistic relit renderings of that object. Of particular interest are renderings of human faces, which have many practical uses within consumer photography and the visual effects industry, but also serve as a particularly challenging case due to their complexity and the high sensitivity of the human visual system to facial appearance. 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 [Debevec et al. 2000].

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, 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. 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. 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. 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) 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. 3.1) and develop a novel alias-free pooling approach to combine their network activations (Sec. 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.

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 [Ramamoorthi and Hanrahan 2001; Sato et al. 2003; Durand et al. 2005], and can also be related to analyses of sampling rate in image-based rendering [Chai et al. 2000] for the related problem of view synthesis [Mildenhall et al. 2019]. This problem also bears some similarities to multi-image super-resolution [Milanfar 2010] and angular super-resolution in the light field [Kalantari et al. 2016; Cheng et al. 2019], where aliased observations are combined to produce interpolated results. In this paper, 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 [Peers et al. 2009; Sen and Darabi 2009], kernel Nystrom [Wang et al. 2009], optical computing [O'Toole and Kutulakos 2010] and neural networks [Ren et al. 2013, 2015; Kang et al. 2018]. 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, [Xu et al. 2018] 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 [Woodham 1980], 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 [Tunwattanapong et al. 2013]. There are also works that focus on recovering a parametric model from a single image [Sengupta et al. 2018], constructing a volumetric model for view synthesis [Lombardi et al. 2018], or even a neural representation of a scene [Tewari et al. 2020]. 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 [Tunwattanapong et al. 2011; Debevec 2012].

Interpolating a reflectance function has also been investigated in the literature. Masselus et al. [2004] 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. [2019] 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. [2007] 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 [Meka et al. 2019; Guo et al. 2019; Sun et al. 2019; Nestmeyer et al. 2019]. 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 [Sun et al. 2019; Apple 2017] or VR experiences [Guo et al. 2019]. These learning-based applications suffer from the same undersampling issue as do conventional uses of light stage data. For example, Sun et al. [2019] observe artifacts when relighting with environment maps that contain high-frequency illumination. We believe our method can provide better training data and significantly improve many of these methods in the future.

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. 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 $\boldsymbol{\ell}$ and a complete OLAT scan consisting of a set of paired images and light directions $\{\mathbf{I}_i, \boldsymbol{\ell}_i\}$, and uses a deep neural network Φ to obtain the predicted image $\mathbf{I}$,

$$\mathbf{I}(\boldsymbol{\ell}) = \Phi\left(\{\mathbf{I}_i, \boldsymbol{\ell}_i\}_{i=1}^{n}, \boldsymbol{\ell}\right). \tag{1}$$

This formalization is broad enough to describe some prior works on learning-based relighting [Xu et al. 2018; Meka et al. 2019]. While these methods usually operate by training a U-Net [Ronneberger et al. 2015] 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 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. 2. Given a query light direction $\boldsymbol{\ell}$, 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* $\mathrm{A}(\boldsymbol{\ell})$. These OLAT images $\mathbf{I}_i$ and their corresponding light directions $\boldsymbol{\ell}_i$ 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 $\mathrm{W}(\boldsymbol{\ell}, \boldsymbol{\ell}_i)$. 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 $\boldsymbol{\ell}$, 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 $\mathbf{I}(\boldsymbol{\ell})$. Formally, our final image synthesis procedure

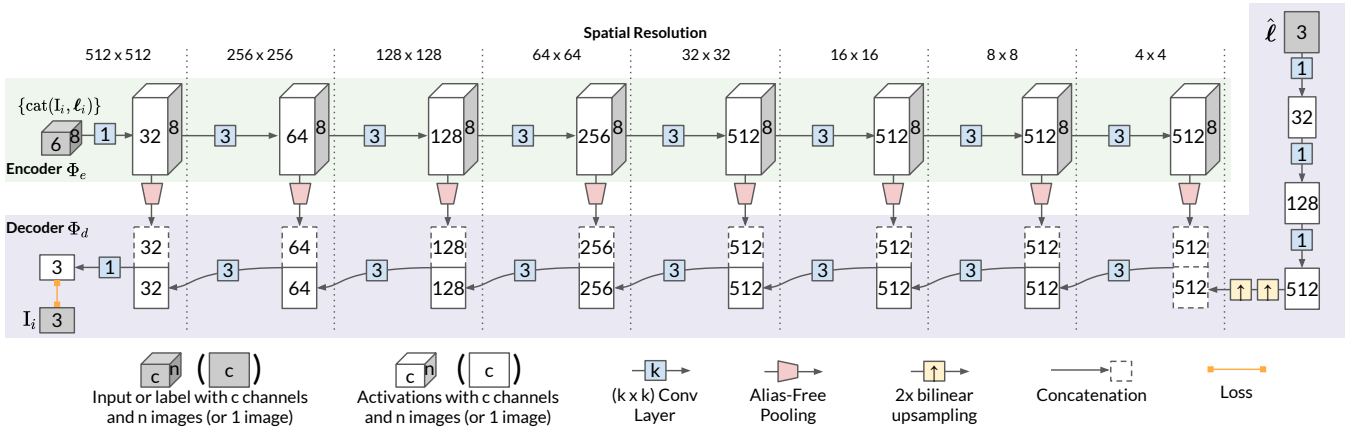


Fig. 2. A visualization of our model architecture. The encoder of our model $\Phi_e(\cdot)$ takes as input a concatenation of the nearby OLAT images in the active set and their light directions, which are processed by a series of stride-2 conv layers. The resulting encoded activations of these 8 images at each level are then combined using the alias-free pooling described in Section 3.2, and skip-connected to the decoder. The decoder $\Phi_d(\cdot)$ takes as input the query light direction $\boldsymbol{\ell}$, 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.

is:

$$\mathbf{I}(\boldsymbol{\ell}) = \Phi_d\left(\sum_{i \in \mathrm{A}(\boldsymbol{\ell})} \mathrm{W}(\boldsymbol{\ell}, \boldsymbol{\ell}_i)\Phi_e\left(\mathbf{I}_i, \boldsymbol{\ell}_i\right), \boldsymbol{\ell}\right). \tag{2}$$

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 3.1, our alias-free pooling is explained in Section 3.2, the network architecture is described in Section 3.3, and our progressive training procedure is discussed in Section 3.4.

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. 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 be a problem when evaluating our model, as will be discussed in Sec. 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. 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. 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.

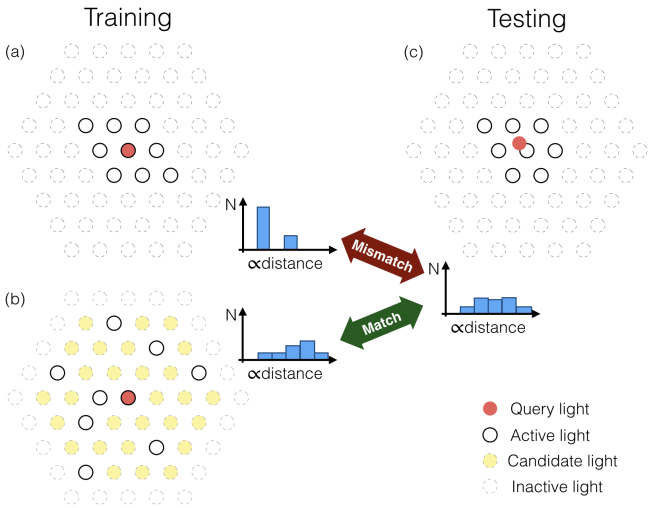


Fig. 3. The OLAT images taken from a light stage have a uniform hexagonal pattern, which means that the distances between each light and its nearest neighbors is highly regular (a). In contrast, at test time we want to synthesize images corresponding to unseen light directions that do not lie on this 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.

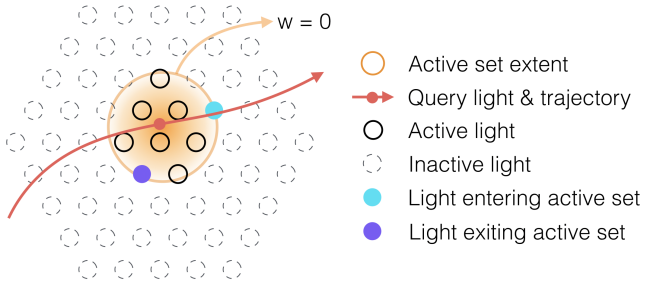


Fig. 4. Varying the query light direction will cause OLAT images to leave and enter the active set of our model, which introduces aliasing that, if unaddressed, results in jarring temporal artifacts in our renderings. To address this, we use an "alias-free pooling" technique to ensure that the network 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.

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 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. 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 [Srivastava et al. 2014] 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. 4 and in the supplementary video, 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.

3.2 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. 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 $\boldsymbol{\ell}$ and each OLAT light direction $\boldsymbol{\ell}_i$ as follows:

$$\begin{aligned} \widetilde{\mathrm{W}}(\boldsymbol{\ell}, \boldsymbol{\ell}_i) &= \max\left(0, e^{s(\boldsymbol{\ell}\cdot\boldsymbol{\ell}_i - 1)} - \min_{j \in \mathrm{A}(\boldsymbol{\ell})} e^{s(\boldsymbol{\ell}\cdot\boldsymbol{\ell}_j - 1)}\right), \\ \mathrm{W}(\boldsymbol{\ell}, \boldsymbol{\ell}_i) &= \frac{\widetilde{\mathrm{W}}(\boldsymbol{\ell}, \boldsymbol{\ell}_i)}{\sum_j \widetilde{\mathrm{W}}(\boldsymbol{\ell}, \boldsymbol{\ell}_j)}, \end{aligned} \tag{3}$$

where s is a learnable parameter that adjusts the decay of the weight with respect to the distance and each $\boldsymbol{\ell}$ 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 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. 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 [Wu and He 2018] and a PReLU [He et al. 2015] 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 3.2, 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 $\boldsymbol{\ell}$, each of which is followed by instance normalization [Ulyanov et al. 2016] 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 [Long et al. 2015], 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.

3.4 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:

$$\mathcal{L}_d = \sum_i \left\| \mathbf{M} \odot (\mathbf{I}_i - \mathbf{I}(\boldsymbol{\ell}_i)) \right\|_1, \tag{4}$$

where $\mathbf{I}_i$ is the ground truth image under light i, and $\mathbf{I}(\boldsymbol{\ell}_i)$ 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 $\boldsymbol{\ell}_i$. We then identify a set of 8 neighboring images/lights to include in our active set using the selection procedure described in Section 3.1. Our only data augmentation is a randomly-positioned 512×512 crop of all images in each batch.

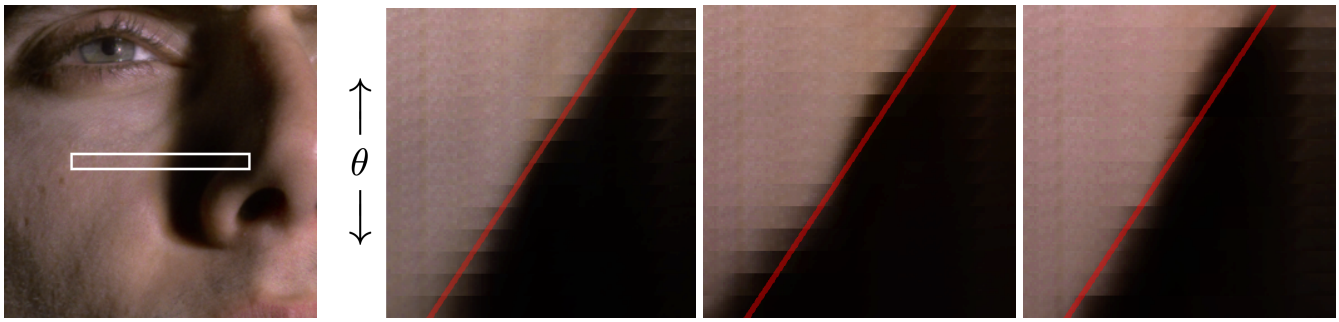
Progressive training has been found to be effective for accelerating and stabilizing the training of GANs for high-resolution image synthesis [Karras et al. 2018], and though our model is not a GAN (but is instead a convolutional encoder-decoder architecture with skip connections) we found it to also 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. Please see the detailed training procedure in the supplementary material.

Our model is implemented in Tensorflow [Abadi et al. 2016] and trained using Adam [Kingma and Ba 2015] 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 ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-7}$).

4 EVALUATION

We use the OLAT portrait dataset from [Sun et al. 2019], 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 [Wenger et al. 2005] 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 accompanying 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 3.1, 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



(a) One rendering, for reference (b) Our model (c) Our model w/ naive neighbors (d) Our model w/ avg pooling

Fig. 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 $\boldsymbol{\ell}$ 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.

Table 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. 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 et al. [2007]	0.0195	0.0258	0.0382	0.00485
Photometric stereo	0.0284	0.0362	0.0968	0.00895
Xu et al. [2018]				
w/ 8 optimal lights	0.0410	0.0437	0.1262	0.01666
w/ adaptive input	0.0259	0.0291	0.1156	0.00916
Meka et al. [2019]	0.0505	0.0561	0.1308	0.01482

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 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 [Ng et al. 2003], DSSIM [Wang et al. 2004], and E-LPIPS [Kettunen et al. 2019]. 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.

4.1 Ablation Study

We first evaluate against ablated versions of our model, with results shown in Tab. 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. 1) but also significant overfitting: this apparent improvement in performance is misleading, because the validation set of our dataset 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. 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. See the supplemental video for additional visualizations.

In the "Ours w/avg-pooling" ablation we replace the alias-free pooling of our model with simple average pooling. As shown in Tab 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. See the supplemental video for examples.

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. 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

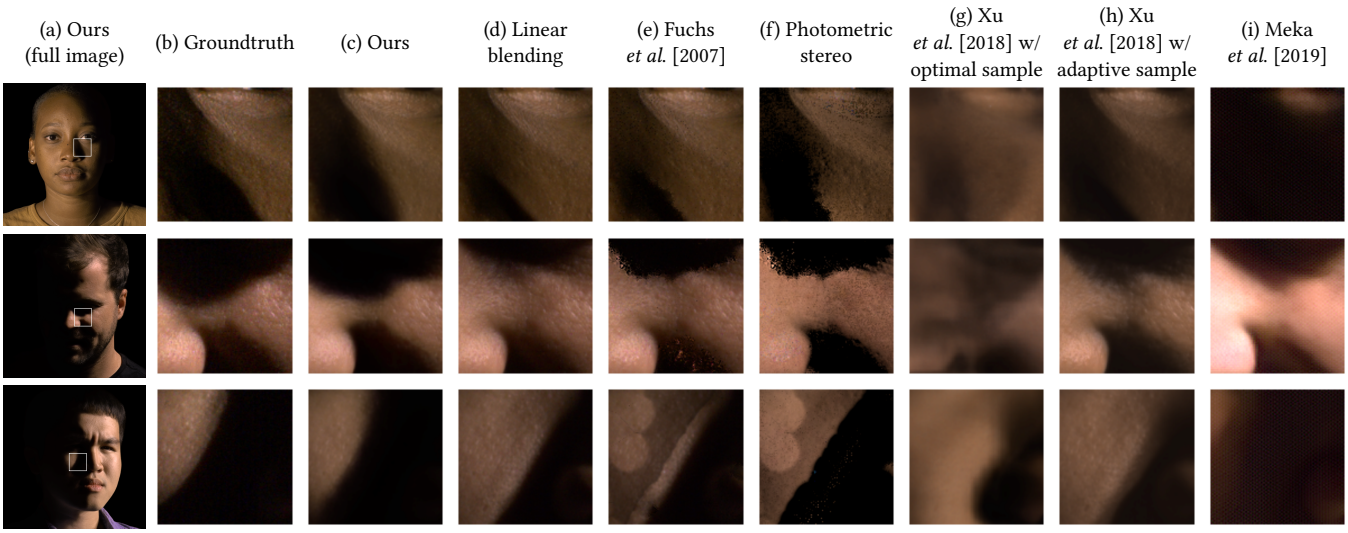


Fig. 6. Here we present a qualitative comparison between our method and other light interpolation algorithms. Traditional methods (linear blending, Fuchs et al. [2007], photometric stereo) retain detail but suffer from ghosting artifacts in shadowed regions. Results from Xu et al. [2018] and Meka et al. [2019] exhibit significant oversmoothing and brightness changes. Our method retains details and synthesizes shadows that resemble the ground truth.

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. 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. [2007] 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. 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. [2007]. 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. 6 and as reflected by the E-LPIPS metric.

Using the layer decomposition produced by Fuchs et al. [2007], 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 [Fuchs et al. 2007] to produce our final rendering. As shown in Tab. 1, this approach underperforms that of Fuchs et al. [2007], 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. 6. In contrast to this photometric stereo approach and the layer-based approach of Fuchs et al. [2007], 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. [2018] (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× 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. [2018]. Table 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. [2018] 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. 3.1) this baseline is able to better reason about local information, which improves performance as shown in Tab. 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 [Meka et al. 2019] for our task, which is also outperformed by our model. This is likely

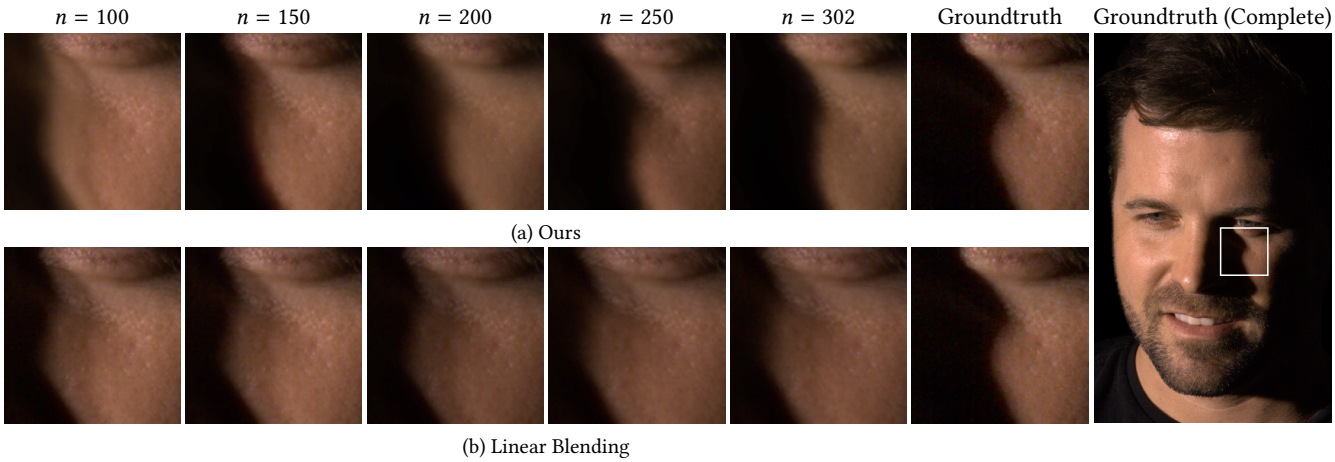


Fig. 7. Here we compare the performance of our model against linear blending as we reduce n, the number of lights in our light stage. As we decrease the number of available lights from $n = 302$ to $n = 100$, the quality of our model's rendered shadow degrades slowly. Linear blending, in contrast, is unable to produce an accurate rendering even with access to all lights.

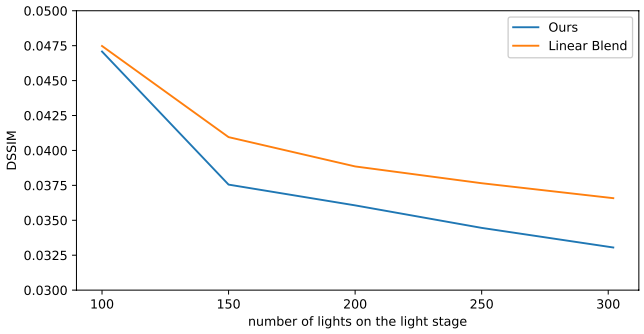


Fig. 8. The image quality of relighting algorithms will gradually reduce as we remove lights from the light stage. However, our algorithm is able to retain the image quality to a greater extent with fewer lights compared to naive linear blending.

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.

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. 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. 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 shadows and sharp highlights. Please refer to our supplementary video for our qualitative comparison.

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. 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. [2018] with adaptive sampling does not produce realistic results and always contains multiple superposed shadows or highlights. The shadows produced by Meka et al. [2019] are sharp, but are not moving smoothly when the 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.

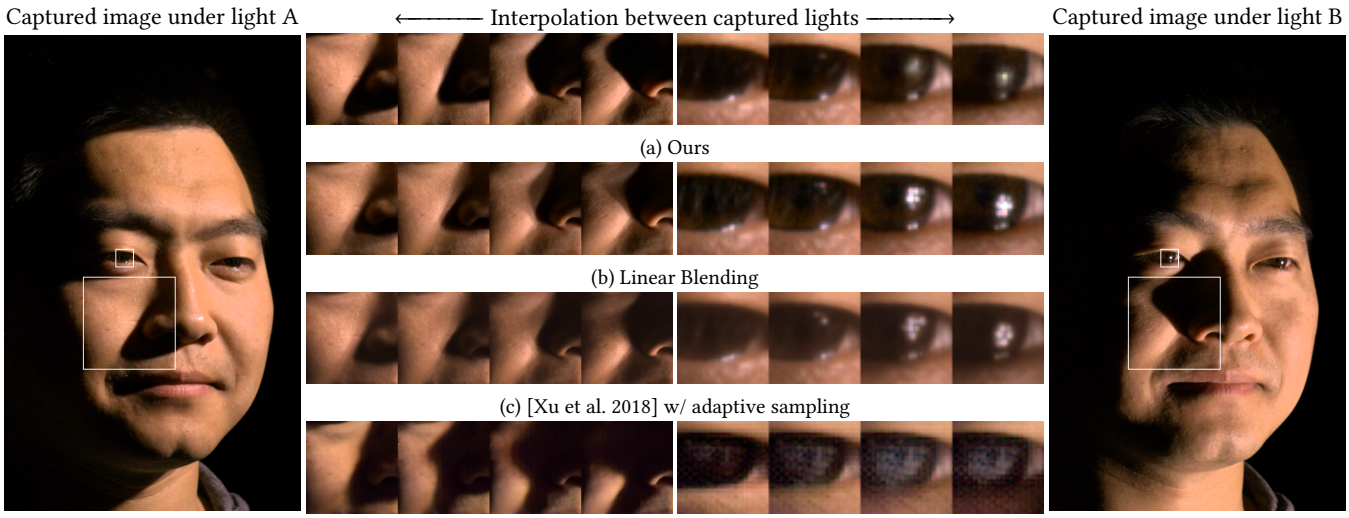


Fig. 9. Here we use produce interpolated images corresponding to "virtual" lights between two real lights of the light stage. Our model (a) produces renderings where sharp shadows and accurate highlights move realistically. Linear blending (b) and Xu et al. [2018] with adaptive sampling result in ghosting artifacts and duplicated highlights. The results from Meka et al. [2019] contain blurry highlights and shadows with unrealistic motion.

(a) With super-resolution (b) Without super-resolution

Fig. 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 [Debevec et al. 2000]. Using the light stage data as-provided (b) results in ghosting.

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 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. 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. Please see our video for more environment relighting results.

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 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. 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

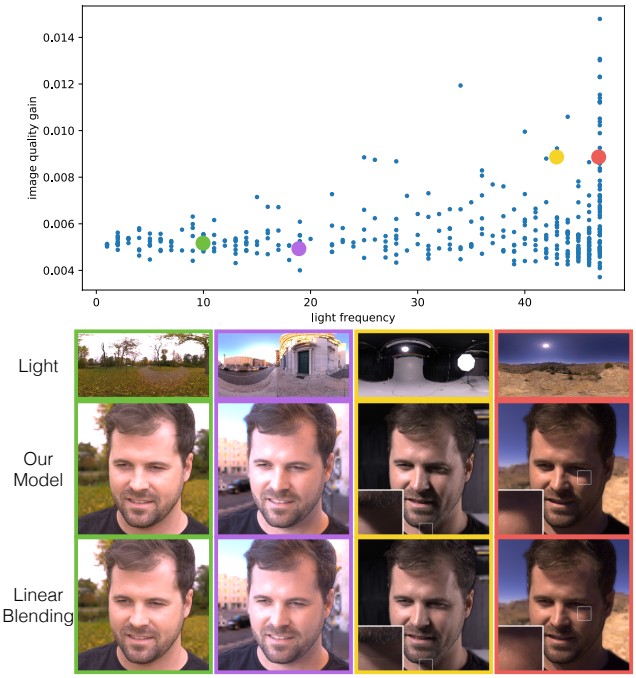


Fig. 11. In the top figure, each blue dot represents a lighting environment. We render a portrait under this environment using both linear-blending and our method, and measure the image difference using SSIM to evaluate the quality gain of our algorithm. The image quality improvement produced by our 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).

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. 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

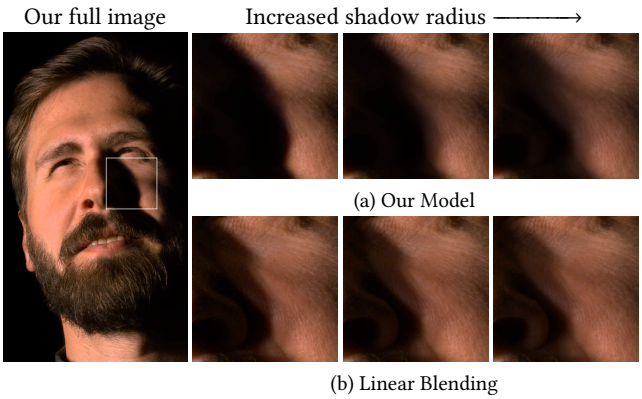


Fig. 12. Soft shadows can be rendered by synthesizing and averaging images corresponding to directional light sources within some area on the sphere. Soft shadows rendered by our method (a) are more realistic and contain fewer ghosting artifacts than those rendered using linear blending (b).

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. 12, our model is able to synthesize realistic shadows with controllable softness, which is not possible using traditional linear blending methods.

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. 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.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants 1617234, 1703957 ONR grants N000141712687 and N000142012529, a Google Fellowship, the Ronald L. Graham Chair, and the UC San Diego Center for Visual Computing. Thanks to anonymous reviewers for the valuable feedback.

REFERENCES

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, et al. 2016. TensorFlow: A system for large-scale machine learning. *OSDI* (2016).

Apple. 2017. Use Portrait mode on your iPhone. https://support.apple.com/en-us/HT208118.

Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. 2000. Plenoptic Sampling. In *SIGGRAPH*.

Zhen Cheng, Zhiwei Xiong, Chang Chen, and Dong Liu. 2019. Light Field Super-Resolution: A Benchmark. In *CVPR Workshops*.

Paul Debevec. 2012. The Light Stages and Their Applications to Photoreal Digital Actors. In *SIGGRAPH Asia*.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *SIGGRAPH*.

Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. 2005. A Frequency Analysis of Light Transport. In *SIGGRAPH*.

Martin Fuchs, Hendrik PA Lensch, Volker Blanz, and Hans-Peter Seidel. 2007. Super-resolution reflectance fields: Synthesizing images for intermediate light directions. In *Computer Graphics Forum*, Vol. 26. Wiley Online Library, 447–456.

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. 2019. The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting. In *SIGGRAPH Asia*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CVPR* (2015).

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-based view synthesis for light field cameras. *SIGGRAPH* (2016).

Kaizhang Kang, Zimin Chen, Jiaping Wang, Kun Zhou, and Hongzhi Wu. 2018. Efficient reflectance capture using an autoencoder. *SIGGRAPH* (2018).

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *ICLR*.

Markus Kettunen, Erik Härkönen, and Jaakko Lehtinen. 2019. E-LPIPS: Robust Perceptual Image Similarity via Random Transformation Ensembles. *CoRR* abs/1906.03973 (2019). http://arxiv.org/abs/1906.03973

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *ICLR* (2015).

Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep appearance models for face rendering. *SIGGRAPH* (2018).

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Vincent Masselus, Pieter Peers, Philip Dutré, and Yves D Willemsy. 2004. Smooth reconstruction and compact representation of reflectance functions for image-based relighting. In *Proceedings of the fifteenth eurographics conference on rendering techniques*.

Abhimitra Meka, Christian Haene, Rohit Pandey, Michael Zollhöfer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, et al. 2019. Deep Reflectance Fields: High-Quality Facial Reflectance Field Inference from Color Gradient Illumination.

Peyman Milanfar. 2010. *Super-resolution imaging*. CRC Press.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. In *SIGGRAPH*.

Thomas Nestmeyer, Iain Matthews, Jean-François Lalonde, and Andreas M Lehrmann. 2019. Structural Decompositions for End-to-End Relighting. *arXiv preprint arXiv:1906.03355* (2019).

Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. 2003. All-Frequency Shadows using Non-Linear Wavelet Lighting Approximation. In *SIGGRAPH*.

Matthew O'Toole and Kiriakos N. Kutulakos. 2010. Optical Computing for Fast Light Transport. In *SIGGRAPH*.

Pieter Peers, Dhruv K Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. 2009. Compressive Light Transport Sensing. *ACM TOG* (2009).

Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural btf compression and interpolation. In *Computer Graphics Forum*.

Ravi Ramamoorthi and Pat Hanrahan. 2001. A Signal-Processing Framework for Inverse Rendering. In *SIGGRAPH*.

Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. 2015. Image Based Relighting Using Neural Networks. *ACM TOG* (2015).

Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. 2013. Global illumination with radiance regression functions. *SIGGRAPH* (2013).

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*.

I Sato, T Okabe, Y Sato, and K Ikeuchi. 2003. Appearance Sampling for Obtaining a set of basis images for variable illumination. In *ICCV*.

P. Sen and S. Darabi. 2009. Compressive Dual Photography. *Computer Graphics Forum* (2009).

Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. 2018. SfSNet: Learning Shape, Refectance and Illuminance of Faces in the Wild. In *CVPR*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a Simple Way to Prevent Neural Networks from Overfitting. *JMLR* (2014).

Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E. Debevec, and Ravi Ramamoorthi. 2019. Single Image Portrait Relighting. *SIGGRAPH* (2019).

Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. 2020. State of the Art on Neural Rendering. (2020).

Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. 2013. Acquiring reflectance and shape from continuous spherical harmonic illumination. *SIGGRAPH* (2013).

Borom Tunwattanapong, Abhijeet Ghosh, and Paul Debevec. 2011. Practical image-based relighting and editing with spherical-harmonics and local lights. In *2011 Conference for Visual Media Production*. IEEE.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).

J Wang, Y Dong, X Tong, Z Lin, and B Guo. 2009. Kernel Nystrom method for light transport. *ACM Transactions on Graphics* (2009).

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *TIP* (2004).

Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. 2005. Performance Relighting and Reflectance Transformation with Time-multiplexed Illumination. *SIGGRAPH* (2005).

Robert J. Woodham. 1980. Photometric Method For Determining Surface Orientation From Multiple Images. *Optical Engineering* (1980).

Yuxin Wu and Kaiming He. 2018. Group Normalization. In *ECCV*.

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. 2018. Deep image-based relighting from optimal sparse samples. In *SIGGRAPH*.