

A Light Stage on Every Desk

Soumyadip Sengupta

Brian Curless

Ira Kemelmacher-Shlizerman

Steve Seitz

University of Washington

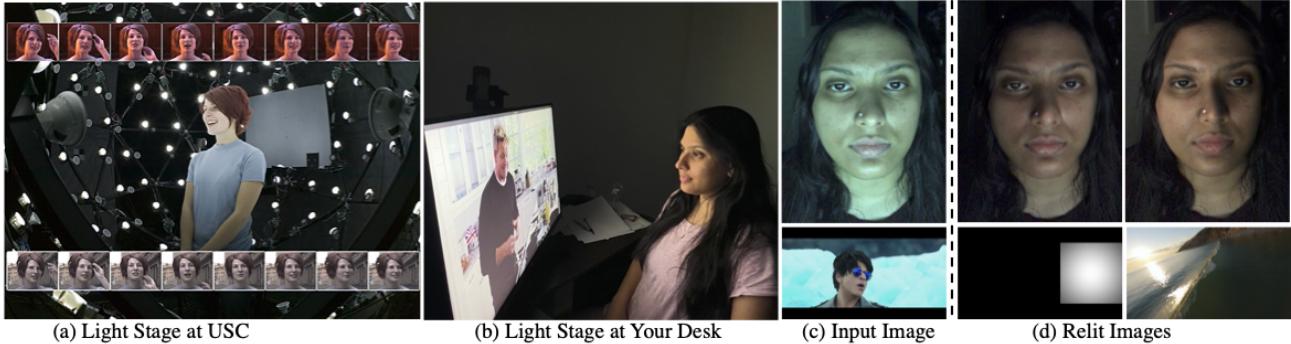


Figure 1: (a) A light stage is a large spherical rig used to capture and relight subjects using multiple lights and cameras [21]. (b) We show how any monitor can be used as a light stage by capturing a user’s facial appearance as they watch a YouTube video to learn a person specific relighting model. Webcam is on top of the monitor. (c) Captured and (d) synthetically re-lit image with previously unseen lighting.

Abstract

Every time you sit in front of a TV or monitor, your face is actively illuminated by time-varying patterns of light. This paper proposes to use this time-varying illumination for synthetic relighting of your face with any new illumination condition. In doing so, we take inspiration from the light stage work of Debevec et al. [4], who first demonstrated the ability to relight people captured in a controlled lighting environment. Whereas existing light stages require expensive, room-scale spherical capture gantries and exist in only a few labs in the world, we demonstrate how to acquire useful data from a normal TV or desktop monitor. Instead of subjecting the user to uncomfortable rapidly flashing light patterns, we operate on images of the user watching a YouTube video or other standard content. We train a deep network on images plus monitor patterns of a given user and learn to predict images of that user under any target illumination (monitor pattern). Experimental evaluation shows that our method produces realistic relighting results. Video results are available at grail.cs.washington.edu/projects/Light_Stage_on_Every_Desk/.

1. Introduction

A light stage, first introduced by Debevec *et al.* [4], is an instrument to capture and render human subjects under almost any illumination condition. Most light stages consist of room-scale, spherical arrays of brightly-flashing colored

lights and cameras. They are used widely for movie special effects [5, 2, 1, 21, 15], volumetric media [25, 6], presidential portraits [12], and to provide rich data for training computer vision relighting algorithms [18, 24, 19, 9, 13, 20, 10].

Unfortunately, very few researchers have access to a light stage, as only a few exist in the world. Furthermore, light stage datasets (necessary to train relighting algorithms) are not publicly available. The goal of this paper is to democratize light stage capture and the development of new relighting algorithms.

Our key insight is that the simple act of watching video on a monitor or TV sends patterns of light across your face. By analyzing these patterns, we can achieve face relighting in a manner similar to a light stage. The ability to capture “light stage” data from displays we use everyday and from the content we already watch dramatically broadens the access and applicability of light stage and relighting research. For example, imagine improving how your face is lit in a video call, by analyzing the changes in your facial appearance via monitor lighting over the last 10 minutes. However, it comes with some new challenges, namely:

1. *passive vs. active lighting*: we wish to avoid forcing the user to watch flashing light patterns, and instead operate on the natural, time-varying content people normally watch.
2. *motion*: user head motion combined with time-varying light patterns complicate the registration problem (in contrast to traditional light stage techniques, which send white frames to enable optical flow).

3. *field of view*: the monitor provides only frontal lighting from a limited field of view.
4. *brightness*: we are limited by monitor brightness relative to room lighting.
5. *near-field*: the monitor is a proximal source, and does not model distant lighting.

In this paper, we address the first two of these challenges, which effectively transform every desktop monitor into a light stage. We show that the resulting data enables new relighting algorithms and applications such as improving face lighting for video calls. While we believe future research can improve the operating range, our approach works best for dimly lit rooms or bright monitors.

To this end, we propose a deep network that takes as input a face image and corresponding source pattern (image on monitor), and produces an image of the same face under a desired target pattern. This network is trained using imagery of a moving person watching a (known) monitor video. To normalize for user motion, we find pairs of images where the face is approximately in the same pose, but with different monitor patterns. We then train the network to relight the first image of the pair with the pattern of the other, using perceptual, cycle consistency and adversarial losses based on PatchGAN [27]. We handle lighting as a style, and use Adaptive Instance Normalization (AdaIN) [7] to infuse lighting information into our deep network. The parameters of AdaIN are learnt from input and target lighting pattern with a multi-layer perceptron. We show state-of-the-art results on a variety of scenes and captures.

In the following section we define the problem and reference related work, as well as fold related work into the subsequent sections.

2. Problem Definition

The *performance relighting* problem [21] is to capture video of a moving subject that can be synthetically re-lit with new target lighting. Specifically, given a set of input images of the subject $\mathbf{I} = [I^1 \dots I^N]$, input light maps $\mathbf{L} = [L^1 \dots L^N]$, and target light maps $\mathbf{L}_t = [L_t^1 \dots L_t^N]$, predict how each input image I^i appears under target L_t^i , to produce re-lit image sequence $\mathbf{I}_t = [I_t^1 \dots I_t^N]$. (The representation of L^i is implementation-dependent.) While performance relighting is limited to re-illuminating the specific person who was captured, many solutions enable very detailed, view-dependent effects, and movie-quality output.

In contrast, *portrait relighting* methods operate on a *single image of a subject* I , transforming it to a re-lit image I_t under a target light L_t . While earlier deep net methods leveraged synthetic data [17, 26], most state-of-the-art techniques now rely on light stage data for portrait relighting [18, 24, 19, 9, 13, 20, 10]. Portrait relighting methods require training data spanning a wide range of people.

In this paper, we focus on the performance relighting

problem where the illumination comes from a conventional monitor or TV. Prior research has also considered using monitors [3, 28] or projection screens [16] as light sources to enable relighting of static objects. In contrast, our approach is suitable for relighting videos, for example to improve facial appearance in YouTube or video calls. A second application is to normalize facial lighting for game players or other users in front of monitors. As inverse lighting methods progress, it may become possible to decode screen content from face video [14]; performance relighting provides a way to foil such privacy attacks.

3. Monitor as a Light Stage

In our setup, the subject sits in front of a monitor while being passively (and primarily) illuminated by screen content, e.g., a YouTube video. Figure 1(b) illustrates this simple scenario. We record both the frames shown on the screen and a synchronized webcam view of the subject. The result is a set of screen (lighting) images \mathbf{L} and corresponding images of the subject \mathbf{I} .

Later, while still capturing the screen and subject, we seek to relight the subject. This means at test time, we have source lighting L_s , source image I_s , and target lighting L_t , and seek to produce relit image \hat{I}_t .

3.1. Linear baseline

At first glance the problem and solution seem straightforward. Indeed, if the subject is perfectly still and we follow the light stage approach of illuminating one pixel (or non-overlapping group of pixels) at a time, then we can simply solve for the weighted combination of input lights that produce the target lighting:

$$L_t = \sum_i w_i L^i = \mathbf{L}w, \quad (1)$$

where w is the weight vector $[w_1 \dots w_N]^T$.

Given the linearity of incoherent light transport (and after linearizing for camera response), we can apply those same weights to the input images to get the relit image:

$$\hat{I}_t = \sum_i w_i I^i = \mathbf{I}w. \quad (2)$$

Solving for w is trivial if pixel groups are turned on one at time (\mathbf{L} is identity matrix), but for arbitrary lighting sequences, we can compute the least-squares optimal weights:

$$w = \mathbf{L}^+ L_t \quad (3)$$

where \mathbf{L}^+ is the pseudoinverse of \mathbf{L} . Notice how, in this linear formulation, we don't need the source lighting L_s , or even the source image I_s , to generate the result \hat{I}_t .

The critical assumption in this linear method is that the subject is perfectly still, which is not true in practice. Light stage approaches typically overcome this limitation by asking the subject to hold (fairly) still, or by using a very high

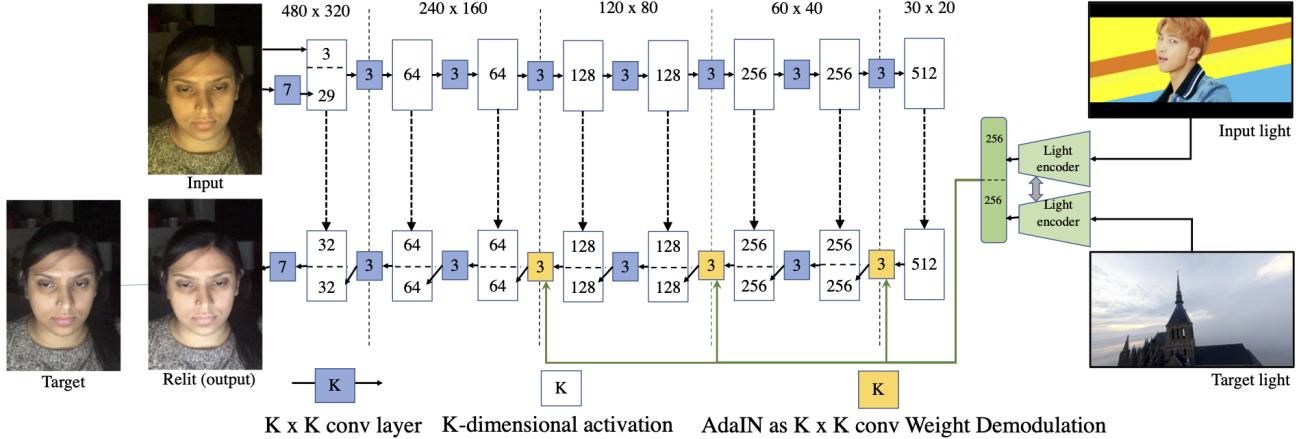


Figure 2: Our proposed architecture consists of an U-Net and a light encoder which estimates a 256 dimensional latent subspace for each of the input and target lights (monitor images). It is then used to demodulate the weights of 3×3 convolutional kernels, similar to Adaptive Instance Normalization (AdaIN) layer, as proposed in StyleGANv2 [8]. The network is trained with L1, perceptual, adversarial, and cycle consistency loss functions.

frame rate camera, and then introducing white frames (all lights on) during capture. This allows optical flow to register all imagery to compensate for small subject motion. In our setting, we work with uncontrolled imagery (no white frames injected) and subjects who move their heads in natural ways (not required to hold still) while watching that imagery in front of a conventional camera.

3.2. Our approach

To address the challenge of subject’s natural head motion while utilizing the lighting and image pairs at test time, we train a deep network G to produce the desired, relit image:

$$\hat{I}_t = G(I_s, L_s, L_t) \quad (4)$$

We take inspiration from single-image portrait relighting techniques, which do not depend on L_s , and generalize to include L_s as input (Section 3.2.1) and handle head motion in the data (Section 3.2.2). The motion is especially challenging because we have a continuous stream of uncontrolled lighting and capture, but no ground-truth supervision in the form of pixel-aligned source and target image pairs, i.e., I_s and I_t . We train instead, on all pairs of images with *roughly similar* poses and design network losses that do not require exact alignment to produce good results. Note that single-image methods [18, 20, 13] are trained on carefully registered data to generalize and relight any image with unknown source lighting. Our approach fits to a single capture of an individual without registered data, but with source lighting available at test time.

3.2.1 Network architecture

We employ a U-net architecture (shown in Figure 2), similar to one used by Sun *et al.* [18]. The U-Net encoder down-samples the image four times, each time by a factor of two, to obtain the latent feature space. The U-Net decoder expands these features with bi-linear upsampling followed by

3×3 convolution. Skip connections from the encoder are also used at each step of the decoder. Activation block involves pixel normalization followed by learnable ReLUs.

Recent state-of-the-art relighting methods have incorporated lighting in different ways. Sun *et al.* [18] simply concatenate the environment map as channels in the latent subspace, while Wang *et al.* [20] modulate the image features at each level of the U-Net decoder producing better results. Our approach is inspired by the latter, though our overall architecture is quite different as Wang *et al.* need normal and albedo maps for training.

As noted earlier, unlike prior single-image relighting methods, we have as input the source lighting L_s , in addition to the target lighting L_t . We employ a single ‘Light Encoder’ (a multi-layer perceptron (MLP)) that maps a lighting image to a low-dimensional code ($d = 256$). We apply the Light Encoder to L_s and to L_t , concatenate the two codes. These codes are then individually used to demodulate the weights of convolutional kernels at all decoder levels, except the highest resolution. MLPs are used for predicting the weights and biases needed for demodulating the convolutional kernels. We empirically observe that this implementation produces less artifacts than using native Adaptive Instance Normalization (AdaIn) [7] layers.

3.2.2 Training

Due to head motion in the dataset, we do not have pixel-aligned images of the subject under different lighting conditions for supervised network training. Instead, we find pairs of images with similar (though not identical) poses. In particular, we apply a face parsing network pre-trained on FFHQ [22] to every image and then find every pair of images with intersection-over-union between the face segments to be greater than 92%. Each pair (I_s, I_t) along with corresponding lighting (L_s, L_t) is used to supervise train-

ing of the network G .

We define a set of losses to encourage the generated image $\hat{I}_t = G(I_s, L_s, L_t)$ to match supervision I_t . First, we define L1 and perceptual losses:

$$\mathcal{L}_{L1} = \|I_t - \hat{I}_t\|_1, \quad (5)$$

$$\mathcal{L}_P = \|F(I_t) - F(\hat{I}_t)\|_2. \quad (6)$$

where, $F(\cdot)$ is a multi-scale, deep feature extractor [23].

For regularization, employ a cycle consistency loss:

$$\mathcal{L}_C = \|I_s - G(\hat{I}_t, L_t, L_s)\|_1. \quad (7)$$

The aforementioned losses often result in blurry re-lit images due to misalignment. Registering the pairs with optical flow is difficult given the lighting differences between them. Similarly, facial keypoint detection is inaccurate in the relatively low-light regime of monitor illumination, making it difficult to obtain accurate warping. To address this blur, we incorporate an adversarial loss in which patches of the relit image \hat{I}_t are compared against patches in I_t . We use the PatchGAN discriminator [27] D which classifies every 70×70 patches as real or fake. We use the LS-GAN [11] framework to train our generator G and discriminator D . For the generator update, we minimize over the parameters of G :

$$\min_G \lambda_{L1}\mathcal{L}_{L1} + \lambda_P\mathcal{L}_P + \lambda_C\mathcal{L}_C + \lambda_D(D(\hat{I}_t) - 1)^2, \quad (8)$$

where each loss depends on $\hat{I}_t = G(\cdot)$. For the discriminator update, we minimize over the parameters of D :

$$\min_D (D(I_t) - 1)^2 + (D(\hat{I}_t))^2. \quad (9)$$

Details. Each input image resolution is 480×320 , and the lighting resolution is 18×32 . The input images are cropped to the subject’s head to limit the effect of the background. We use $\lambda_{L1} = 1$, $\lambda_P = 0.1$, $\lambda_C = 0.5$ and $\lambda_D = 0.1$. We train the generator and discriminator with the Adam optimizer, with learning rate of 10^{-3} and 10^{-6} , and a batch-size of 1.

4. Experimental Evaluation

Capture Our setup consists of a computer monitor and a camera in a dimly lit room (illustrated in Fig 1b). A person watches a YouTube video on the monitor while their appearance is captured by the camera. In our experiments we either turned off the light in the room, or kept low light, and assumed the user is relatively close to the monitor (about $30cm$) to obtain good signal to noise ratio. We tested with both 32” (LG) and 27” (Acer) monitor sizes. The larger monitor is preferred to allow more dramatic illumination effects but the smaller size also works. A 32” monitor provides about 100° horizontal field-of-view and about 70°

vertical field-of-view. For the camera, we used an iPhone 8 mounted with a tripod behind the monitor for some of the experiments, and Samsung S10+ for others, demonstrating that our approach works with different types of cameras and monitors.

We assume the monitor and camera are synchronized, e.g., by flashing an all-white frame at the start and at the end of the captured sequence, and setting both monitor and camera to operate at 30fps. We turn off auto focus and exposure and avoid any post-processing (e.g., night mode) that improves SNR but distorts the synchronous lighting and appearance information. The entire setup was easily replicated by another individual in their home.

Evaluation Protocols To evaluate our approach we capture 19 sequences of 5 individuals watching YouTube videos, and use two protocols for evaluation. Individual A was captured 13 times with 7 different YouTube videos, and Individual B was captured 9 times with 8 different videos. All videos were captured at different days/times, with different ambient lighting and facial conditions. Individual C,D and E was captured once each, where D and E captured themselves in their own home using a 27” monitor and Samsung S10+ camera. Each captured sequence consists of a training video, followed by 2 test videos: 1) moving pattern on the monitor (to test directional lighting) and 2) a random YouTube clip.

The two evaluation protocols are:

Protocol 1 evaluates the typical light stage setting. At test time, we relight a pre-captured user sequence (which is part of the training) with a previously unseen lighting.

Protocol 2 evaluates generalization to unseen instances of a person and lighting conditions during test time. Given multiple captured sequences of a person over a period of time, we train relighting models and test it on a new user sequence with a previously unseen lighting. Note that the facial reflectance of the person during test time can vary from that of the training sequences. This protocol can be useful for relighting a person during video calls.

Baselines Potential baselines for our method are either performance relighting or portrait relighting methods.

Performance relighting typically assumes a linear relighting model combined with optical flow on specific frames captured in the light stage. Since we capture a person’s appearance while watching a regular video, we do not have access to special frames viable for optical flow. The linear model as is (described in section 3.1) fails to produce good results due to natural human motion, thus we omit those from the main paper (see appendix).

Therefore, we focus on portrait relighting as our baseline with the caveat that such single-image methods were developed with a different application in mind. They are trained to work across many individuals, model lighting as a spherical environment map, and require only an image as input.

In contrast we have access to input image and the source lighting, and we model lighting as a video frame.

	Linear model	RSun	Ours
Mean Absolute Error in %	3.37%	3.14%	3.35%

Table 1: Error on mannequin sequence (no misalignment). Our implementation of RSun performs as expected in the ideal case.

Sun *et al.* [18] is a state-of-the-art single image portrait relighting algorithm. It is trained on the OLAT (One Light At a Time) dataset [18] consisting of light stage imagery of multiple individuals. Sun *et al.*'s network consists of an U-Net which predicts both the relit image and the source lighting. It is trained on aligned pair of images with two different lighting conditions, generated from the OLAT data. It uses L1 loss on relit image and L2 loss on predicted lighting, along with a reconstruction loss.

Since neither the data nor model are available we reimplemented the network architecture introduced by Sun *et al.* and train it on the same training data as our algorithm. We use the exact same loss functions proposed in Sun *et al.*. We call this implementation 'RSun'. For perfectly aligned data (mannequin sequence), RSun produced good results, as expected (Table 1). However, RSun performed worse on the rest of our datasets, due likely to the lack of perfect alignment of source and target photos (true for OLAT, but not true for our data). Our data has minor misalignment even after pose matching, which may be causing over-fitting in Sun *et al.* We notice that the reconstruction loss is particularly problematic since during reconstruction path the input and the target is exactly aligned but they are not during the relighting path.

Drawing inspirations from failures of RSun, we introduce a different set of loss functions that can handle the misalignment. We keep the L1 loss on the relit image and L2 loss over predicted lighting and remove the reconstruc-

Algorithm	PSNR (higher is better)	RMSE (lower is better)	LPIPS[23] (lower is better)
RSun	24.51	0.0065	0.2718
Sun+	24.31	0.0067	0.1684
Ours	25.21	0.0054	0.1537

Table 2: We report average PSNR, RMSE and LPIPS [23] score on 19 captured sequences. Protocol 1: Input image appears in training data, target lighting is unseen – light stage.

Algorithm	PSNR (higher is better)	RMSE (lower is better)	LPIPS[23] (lower is better)
Sun+	23.14	0.0064	0.1814
Ours	26.36	0.0034	0.1417

Table 3: We report average PSNR, RMSE and LPIPS [23] score on 8 captured sequences. Protocol 2: input image and target lighting is not part of the training data.

tion loss. Then we add the perceptual loss eqn 6, cycle consistency loss eqn 7, and adversarial loss with discriminator $D(\cdot)$, similar to our approach. This algorithm improves significantly over RSun in many examples. The difference between our approach and Sun+ lies in how lighting is infused into the network architecture. Comparison between RSun, Sun+ and our proposed approach helps us to understand how the loss functions and network architecture both contribute towards improving the quality of the relit results.

4.1. Qualitative and Quantitative Comparison

We present both qualitative and quantitative comparisons. Quantitative evaluation offers some challenges since we can't capture a person in exact same pose with 2 different lighting conditions. At test time, given the target lighting and the face captured with that lighting, we find the input face from the training sequence (in protocol 1), or test sequence (in protocol 2), which has the maximum overlap in face parsing, i.e., most similar pose. Even after finding the nearest pose match, the faces are not exactly the same, defeating measures like RMSE. Instead we use a perceptual measure, LPIPS [23], to evaluate the quality of the relit images with that of the reference image captured under the same lighting. LPIPS also expects aligned images but is more robust to misalignment than RMSE. Since LPIPS with a VGG backbone is used as a loss function, AlexNet backbone is used for evaluation.

Protocol 1: Input image appears in training data, target light is unseen. In Fig 3 we present a qualitative comparison of our approach with that of RSun and Sun+. In Table 2 we present perceptual similarity score LPIPS (lower is better), PSNR (higher is better) and RMSE (lower is better) on all 19 of our captured sequences. Both qualitative in Fig 3 and quantitative evaluation in Table 2 shows that our result is significantly better than RSun and Sun+.

Protocol 2: Input image and target light are not part of training data. Here we consider only individual A and B with 13 and 9 captured sequences each. We then create 4 evaluation setup for each of A and B by leaving 1 sequence out as the test set and considering the rest of the sequences as part of the training set. Test data is captured at different days/times compared to training data, with different YouTube videos. In Table 3 we present quantitative comparison with Sun+ by reporting LPIPS, RMSE and PSNR scores, qualitative evaluation is presented in Fig 4. We show that our approach can also perform relighting by changing light directions and color tones on input images unseen during the training. This means it can be deployed to perform real-time relighting, e.g., during a zoom video call.

Improving lighting for video calls. For the application of relighting a face during a video call, a ring-light pattern is a good choice to generate a well lit face. In Fig 5 we

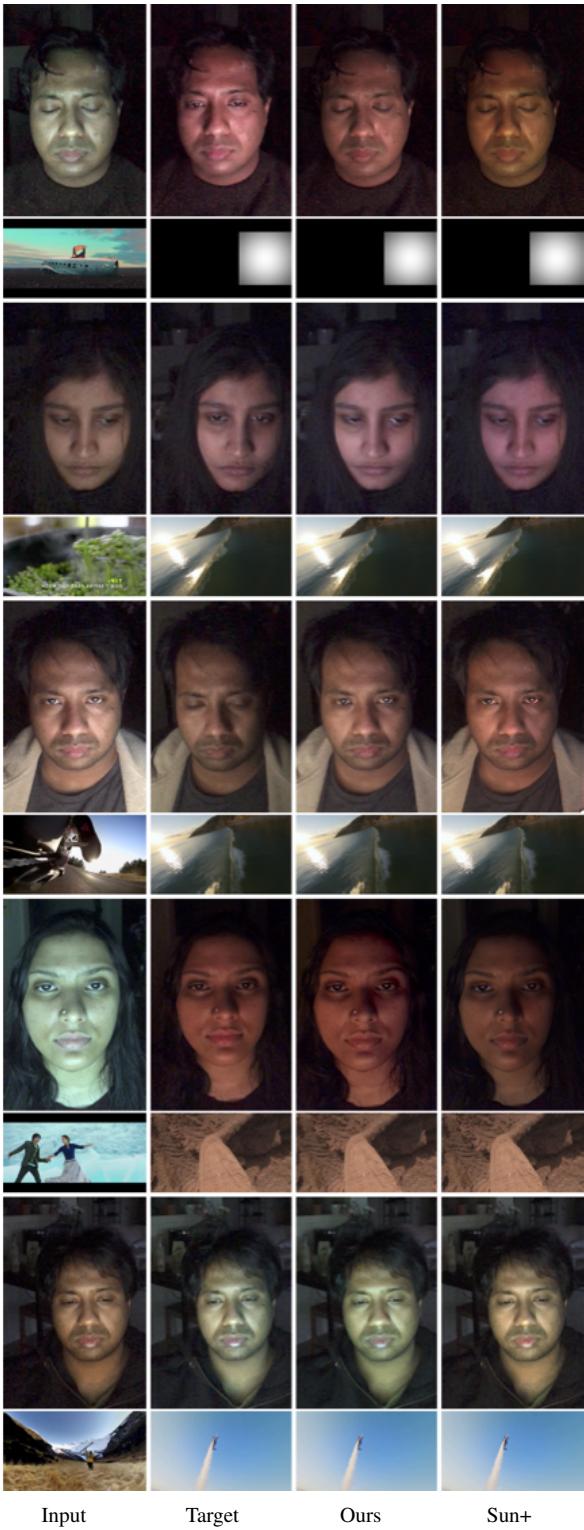


Figure 3: Qualitative comparison with Sun+ following Protocol 1 which resembles light stage testing setup, i.e., input image appears in the training data, while target lighting is unseen.

show some image instances. We observe that the relit im-

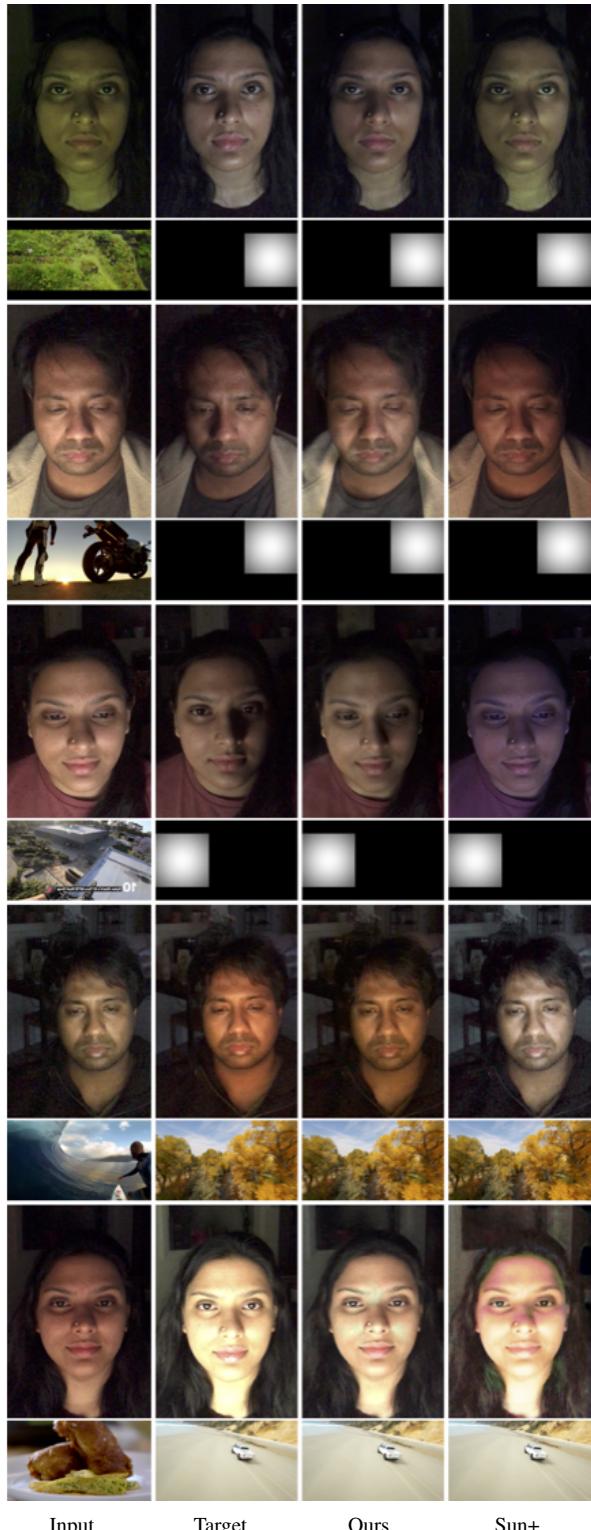


Figure 4: Qualitative comparison with Sun+ following Protocol 2 i.e. input image and target lighting is not part of the training data.

ages remain well-lit and undergo minimal temporal change throughout the video. This follows Protocol 2, i.e. both the

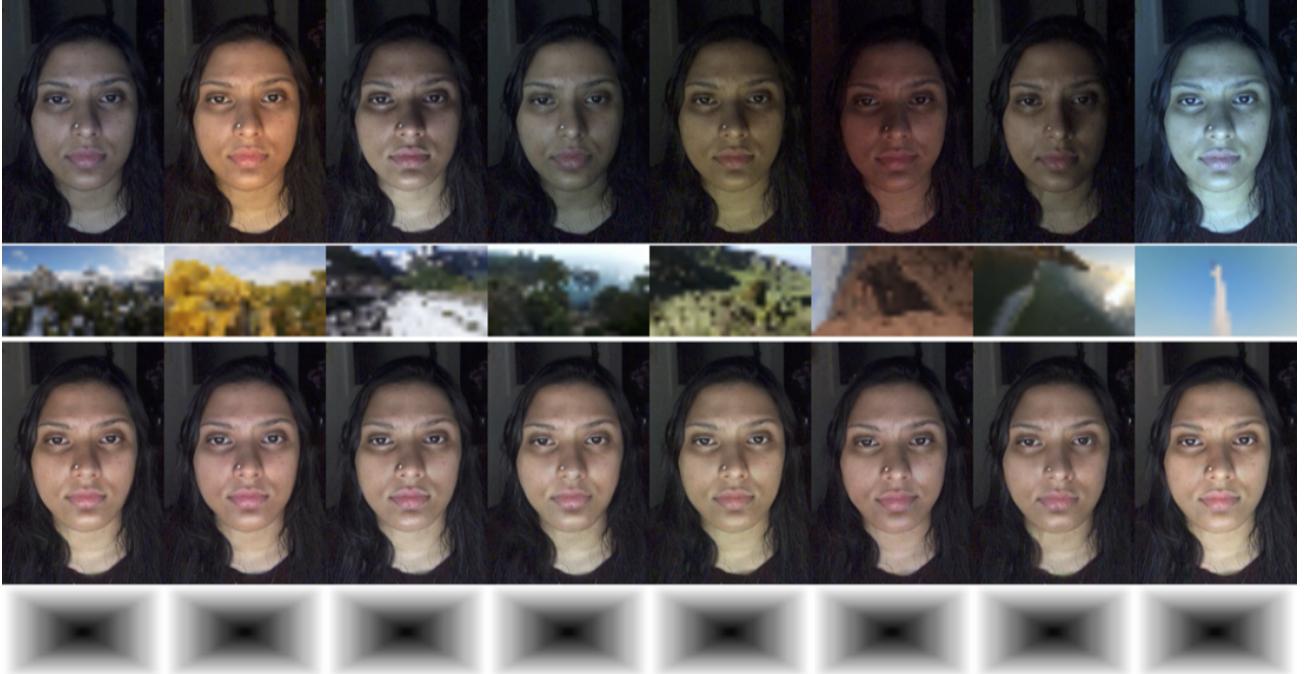


Figure 5: Improving lighting for video calls. During a video call in a poorly-lit room, lighting on the face may change with the content on the monitor (top row). We can re-light the video with a ring-light pattern producing temporally consistent well-lit images (bottom row).

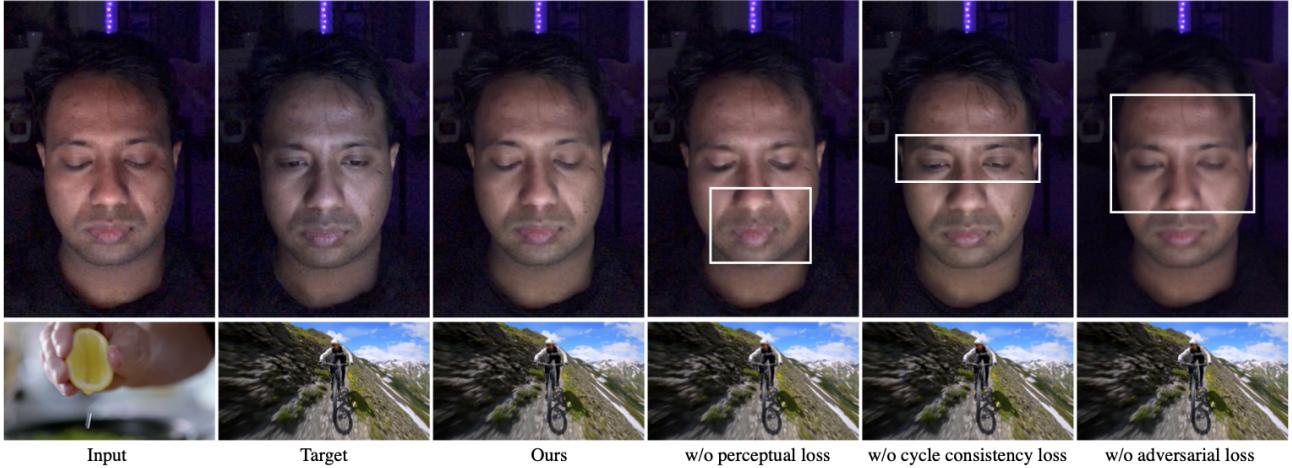


Figure 6: Ablation study w.r.t. loss functions. Both perceptual loss and adversarial loss helps in improving the quality of the relit images, while cycle consistency loss helps in maintaining the exact pose of the input image.

input image and lighting is unseen during the training.

5. Ablation Study

Our combination of data-processing, loss functions and neural architecture helps us to achieve high-quality relighting results from photos captured with monitor emitted illumination. Here we analyse how each component affects the quality of the result. We additionally train our network without using (a) adversarial loss with discriminator $D(\cdot)$, (b) perceptual loss in Eq. 6, and (c) cycle consistency loss in Eq. 7 separately. In Table 4 we compare the effect of

each loss function on 3 captured sequences. Visual results in Figure 6 also shows that the results degrade in absence of perceptual and adversarial losses. On the other-hand in absence of cycle consistency loss, the relighting result changes the expression of the person – eyes change from closed to open. Our network architecture takes both the source and the target monitor lighting as input. We show in Figure 7 that knowing the source monitor lighting can improve the quality of the relit results.

Robustness and Limitations Although our setup operates best in a dark room, we observe it can still handle mod-

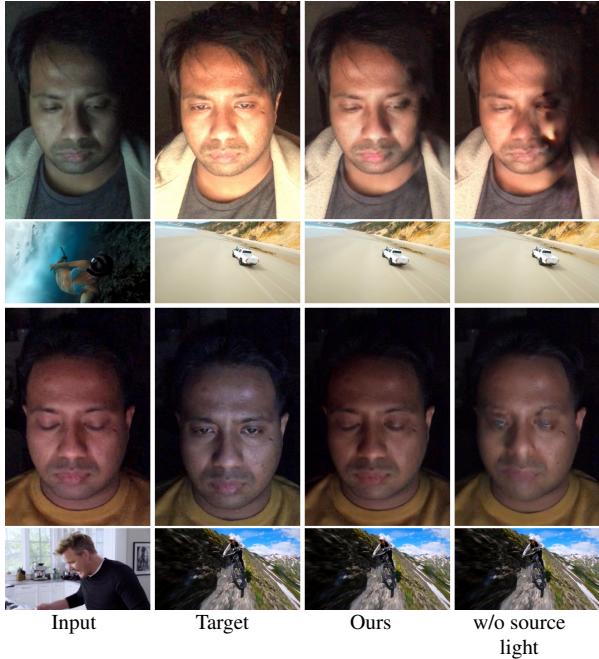


Figure 7: Both the source and target monitor lighting is input to our network. Using source lighting improves the quality.

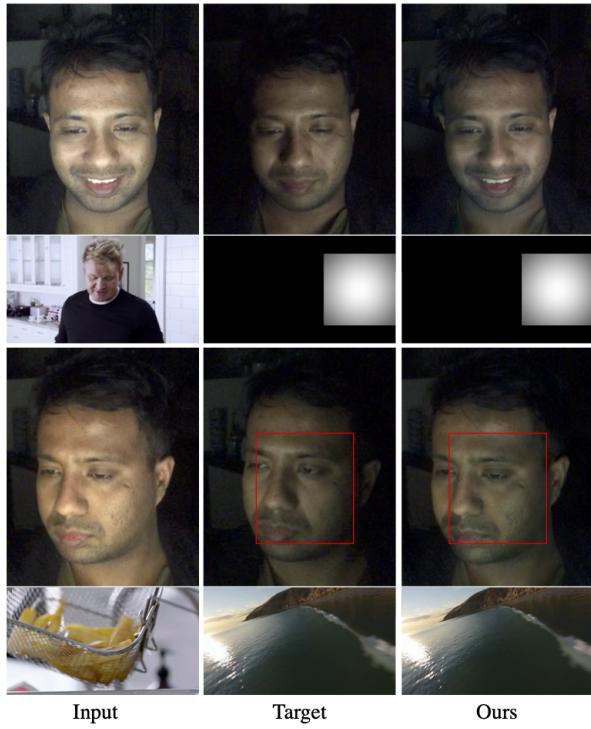


Figure 8: Although our relighting network is trained on nearly frontal faces with neutral expressions, it handles handle unseen expressions like a smiling face during test time (top row). However it is less accurate when the input image has a large pose variation from a frontal face (bottom row).

erate amount of background lighting as long as it is not di-

Algorithm	LPIPS[23] (lower is better)
Ours	0.1311
w/o perceptual loss	0.1384
w/o cycle consistency loss	0.1493
w/o adversarial loss	0.1523
w/o source light	0.1374

Table 4: Ablation study with perceptual similarity LPIPS [23]following Protocol 1 on 3 captured sequence, i.e., input image appears in the training data, while target lighting is unseen.

rected towards the subject. We empirically observe that on average many YouTube videos perform well as a source of lighting. Some categories of videos perform significantly better, e.g., videos with large camera motions, as they provide greater lighting variation. Conversely, videos with limited lighting variations, e.g., a single person talking or dark scenes, do not perform well.

Our method can also handle unseen expressions at test time. In Fig 8 top row, we show an example of an individual smiling while the network was trained on only neutral expressions. However our method is far from perfect in presence of large pose variations in the input image. Cast shadows, while reproduced, can be lower-contrast compared to ground-truth, e.g., around the nose in Fig Fig 8 bottom row. Application of our relighting network during video calls produces mostly uniform and temporally stable results, but can exhibit flickering when the luminance of the input lighting drops significantly (see video results). Adding better alignment across faces, expression normalization, and lighting data augmentation might enable future improvements.

6. Conclusion

This paper introduced a technique to democratize light stage capture for relighting applications, using nothing more than the monitor on your desk. While our approach does not provide the full field of view, dynamic range, and full body coverage of a traditional laboratory light stage, it is easy to deploy and significantly expands access to light stage data and algorithms. Furthermore, we show how relighting models can be trained passively, e.g., from footage of a user watching a normal movie. Our relighting approach is robust to user motion, and produces realistic results.

Ethics. Our primary goal is to improve lighting for videos and video calls. Our approach can also be used to improve privacy, by making it harder to infer screen content reflected from a user’s face. We note, however, that synthetic image relighting is a form of image manipulation, and can facilitate compositing images for negative purposes as well. Forgery detection and prevention is an important and ongoing topic of work.

Acknowledgement This work is supported by the UW Reality Lab, Facebook, Google, Futurewei, and Amazon.

References

- [1] Oleg Alexander, Graham Fyffe, Jay Busch, Xueming Yu, Ryosuke Ichikari, Andrew Jones, Paul Debevec, Jorge Jimenez, Etienne Danvoye, Bernardo Antionazzi, et al. Digital ira: Creating a real-time photoreal digital actor. In *ACM SIGGRAPH 2013 Posters*, pages 1–1. 2013. 1
- [2] Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul Debevec. The digital emily project: Achieving a photorealistic digital actor. *IEEE Computer Graphics and Applications*, 30(4):20–31, 2010. 1
- [3] Yung-Yu Chuang, Douglas E. Zongker, Joel Hindorff, Brian Curless, David H. Salesin, and Richard Szeliski. Environment matting extensions: Towards higher accuracy and real-time capture. In *Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series*, pages 121–130. ACM Press / ACM SIGGRAPH / Addison Wesley Logman, July 2000. ISBN 1-58113-208-5. 2
- [4] 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*, pages 145–156, 2000. 1
- [5] Paul Debevec, Andreas Wenger, Chris Tchou, Andrew Gardner, Jamie Waese, and Tim Hawkins. A lighting reproduction approach to live-action compositing. *ACM Transactions on Graphics (TOG)*, 21(3):547–556, 2002. 1
- [6] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escalano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Paul Debevec, and Shahram Izadi. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Trans. Graph.*, 38(6), Nov. 2019. 1
- [7] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 2, 3
- [8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 3
- [9] Chloe LeGendre, Wan-Chun Ma, Rohit Pandey, Sean Fanello, Christoph Rhemann, Jason Dourgarian, Jay Busch, and Paul Debevec. Learning illumination from diverse portraits. In *SIGGRAPH Asia 2020 Technical Communications*, pages 1–4. 2020. 1, 2
- [10] BR Mallikarjun, Ayush Tewari, Tae-Hyun Oh, Tim Weyrich, Bernd Bickel, Hans-Peter Seidel, Hanspeter Pfister, Wojciech Matusik, Mohamed Elgharib, and Christian Theobalt. Monocular reconstruction of neural face reflectance fields. *arXiv preprint arXiv:2008.10247*, 2020. 1, 2
- [11] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 4
- [12] Adam Metallo, Vincent Rossi, Jonathan Blundell, Günter Waibel, Paul Graham, Graham Fyffe, Xueming Yu, and Paul Debevec. Scanning and printing a 3d portrait of president barack obama. In *SIGGRAPH 2015: Studio*, pages 1–1. 2015. 1
- [13] 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. 1, 2, 3
- [14] Ko Nishino and Shree K. Nayar. Eyes for relighting. *ACM Trans. Graph.*, 23(3):704–711, Aug. 2004. 2
- [15] Pieter Peers, Naoki Tamura, Wojciech Matusik, and Paul Debevec. Post-production facial performance relighting using reflectance transfer. *ACM Transactions on Graphics (TOG)*, 26(3):52–es, 2007. 1
- [16] Yoav Y Schechner, Shree K Nayar, and Peter N Belhumeur. Multiplexing for optimal lighting. *IEEE Transactions on pattern analysis and machine intelligence*, 29(8):1339–1354, 2007. 2
- [17] 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*, pages 6296–6305, 2018. 2
- [18] 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 Trans. Graph.*, 38(4):79–1, 2019. 1, 2, 3, 5
- [19] 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)*, 39(6):1–12, 2020. 1, 2
- [20] Zhibo Wang, Xin Yu, Ming Lu, Quan Wang, Chen Qian, and Feng Xu. Single image portrait relighting via explicit multiple reflectance channel modeling. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020. 1, 2, 3
- [21] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (TOG)*, 24(3):756–764, 2005. 1, 2
- [22] 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. 3
- [23] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4, 5, 8
- [24] Xuaner Zhang, Jonathan T Barron, Yun-Ta Tsai, Rohit Pandey, Xiuming Zhang, Ren Ng, and David E Jacobs. Portrait shadow manipulation. *ACM Transactions on Graphics (TOG)*, 39(4):78–1, 2020. 1, 2

- [25] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escalano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG)*, 40(1):1–17, 2021. [1](#)
- [26] 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. [2](#)
- [27] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. [2, 4](#)
- [28] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In *Proceedings of ACM SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 205–214. ACM Press / ACM SIGGRAPH / Addison Wesley Logman, July 1999. [2](#)