Face Relighting with Geometrically Consistent Shadows

Andrew Hou*,1, Michel Sarkis2, Ning Bi2, Yiyong Tong1, Xiaoming Liu1
1Michigan State University, 2Qualcomm Technologies Inc.
{houandr1, ytong, liuxm}@msu.edu, {msarkis, nbi}@qti.qualcomm.com
https://github.com/andrewhou1/GeomConsistentFR

Abstract

Most face relighting methods are able to handle diffuse shadows, but struggle to handle hard shadows, such as those cast by the nose. Methods that propose techniques for handling hard shadows often do not produce geometrically consistent shadows since they do not directly leverage the estimated face geometry while synthesizing them. We propose a novel differentiable algorithm for synthesizing hard shadows based on ray tracing, which we incorporate into training our face relighting model. Our proposed algorithm directly utilizes the estimated face geometry to synthesize geometrically consistent hard shadows. We demonstrate through quantitative and qualitative experiments on Multi-PIE and FFHQ that our method produces more geometrically consistent shadows than previous face relighting methods while also achieving state-of-the-art face relighting performance under directional lighting. In addition, we demonstrate that our differentiable hard shadow modeling improves the quality of the estimated face geometry over diffuse shading models.

1. Introduction

Single image face relighting is a problem of great interest among the computer vision and computer graphics communities. Relighting consumer photos has been a major driving factor in motivating the problem given widespread interest in photo editing. Face relighting also has applications in other areas such as Augmented Reality (AR) [29], where it can be used to modify facial illuminations to match the environment lighting, and face recognition [16, 33], where it can relight images to frontal illuminations. It is thus relevant both for consumer interests and entertainment and for security applications such as authentication.

Earlier relighting methods [35,43,54] tend to make the simplified assumption that light is naturally scattered by the environment and thus diffuse in nature, and that human skin is a lambertian material. While this is sufficient to model general lighting directions and soft shadows, it does not account for non-lambertian effects such as hard shadows from strong directional lights. This is highly problematic since many light sources in the real world (e.g. the sun) are best modeled as directional lights. In AR/VR, the environment lighting is often also set to be directional lights. In order to enhance photorealism both for in-the-wild consumer photos and in AR/VR, proper hard shadow modeling is a necessity.

One important problem in face relighting is thus handling hard shadows. Most existing methods do not handle non-diffuse lighting and are unable to synthesize realistic hard shadows [35,43,54]. They generally use smooth lighting conditions such as low-order Spherical Harmonics (SH) and train on images with diffuse lighting. While many illumination conditions in the wild are ambient or area-based, these assumptions do not account for the interactions of strong directional lights and point lights, which produce hard shadows. Among current methods that do model hard shadows [10, 29, 30], none are able to guarantee geometrically consistent cast shadows since they do not directly utilize the estimated face geometry to generate them. Without using the geometry directly, the shape of the cast shadows, such as those cast by the nose, may be incorrect.

We introduce a novel differentiable algorithm to estimate
the locations of cast shadows using the principles of ray tracing. We rely on the principle that cast shadows will be located on parts of the face where the projected ray to the light source intersects some occluding surface, such as the nose. Our method can thus leverage the estimated face geometry to produce geometrically consistent hard shadows (see Fig. 1). We further demonstrate that differentiably modeling hard shadows can improve the quality of the face geometry, especially in regions that produce hard shadows (e.g. the nose and near the boundary of the face), compared to models that assume diffuse shading. We therefore show that differentiable hard shadow modeling not only benefits the realism of the relit image, but also the intrinsic component estimation which can benefit other downstream tasks.

Our proposed method thus has four main contributions:

- We propose a single image face relighting method that can produce geometrically consistent hard shadows.
- We introduce a novel differentiable algorithm to estimate facial cast shadows based on the estimated geometry.
- We achieve SoTA relighting performance on 2 benchmarks quantitatively/qualitatively under directional lights.
- Our differentiable hard shadow modeling improves the estimated geometry over models that use diffuse shading.

### 2. Related Work

**Face Relighting** Prior works can be categorized into 4 groups: intrinsic decomposition [2, 3, 5–7, 16–19, 22, 23, 29, 30, 35, 36, 40, 44–47, 51], image-to-image translation [10, 43, 49, 53, 54], style transfer [21, 27, 34, 38, 39], and ratio images [32, 37, 42, 50]. Intrinsic decomposition estimates the geometry, albedo, and lighting and renders the image with a new lighting. Image-to-image translation instead directly estimates the relit image. Style transfer transfers the lighting of the reference image as a style to the input image. Ratio image methods relight by estimating the ratio between the source and target images or illuminations. A summary of our method compared to recent works is shown in Tab. 1.

Nestmeyer et al. [29] model hard shadows using a binary visibility map estimated from a U-Net. The visibility map is not directly constrained by the face geometry and therefore has a large amount of freedom, which can lead to geometrically inconsistent shadows. Also, since it is binary, their cast shadows are black whereas the true intensity of a shadow should match the environment’s ambient light.

Hou et al. [10] utilize shadow masks to assign higher weights along hard shadow borders. While this improves the shadows, they do not use the geometry and thus the shadows can have any shape. Our model produces hard shadows where rays cast from the face to the light source intersect other parts of the face geometry, which ensures that the shadows are geometrically consistent.

Pandey et al. [30] model both diffuse and specular lighting, and can generate non-lambertian effects such as hard shadows. However, they rely on a shading network to produce the relit image given the albedo and light maps, which will have network estimation error. Thus, there is no guarantee that they produce geometrically consistent shadows.

**Differentiable Rendering and Ray Tracing** In recent years, multiple differentiable renderers have been proposed [14, 20, 24, 26, 28] that are suitable for inverse rendering tasks. However, the majority of differentiable renderers do not explicitly model shadows, particularly hard shadows.

The most similar work to ours is Li et al. [20], which proposes a differentiable ray tracer to model hard shadows. Their method operates on meshes and introduces a novel Monte Carlo edge sampling algorithm to handle the non-differentiability along triangle edges. Our shadow modeling is also a form of differentiable ray tracing, but operates on the 2.5D points generated from a face depth map rather than a mesh. Instead of integrating over edges of mesh triangles to determine shading, we find it sufficient to sample points between each point on the face and the light source and assign a cast shadow to the point if one of the sampled points intersects with some facial parts, such as the nose.

Recently, Srinivasan et al. propose NeRV [41], which can model scene-specific hard shadows from any direction more efficiently than prior work. However, they rely on a visibility MLP to predict the locations of shadows, which leaves the possibility of generating geometrically inconsistent shadows due to network estimation error. Furthermore, NeRV requires hundreds of images with known lighting and pose and can only be trained on one static scene at a time, whereas our model is generalizable due to leveraging public face datasets in training, only requires a single image per subject, and can be applied in inference to any face image.

### 3. Proposed Method

#### 3.1. Problem Formulation

Our relighting method relies on intrinsic decomposition, and is thus motivated by the rendering equation [11]:

$$L_o(x, \omega_o) = \int_{\omega_i \in \Omega} f(x, \omega_i, \omega_o) L_i(x, \omega_i) \langle n, \omega_i \rangle d\omega_i,$$

$$L_i(x, \omega_i) = \int_{\omega_o} h(x, x', \omega_o, \omega_i) L_o(x', \omega_o) \langle n, \omega_o \rangle d\omega_o$$

### Table 1. Method Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Lighting Model</th>
<th>Model Category</th>
<th>Handles Hard Shadows</th>
<th>Geom. Consistent Hard Shadows</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISNet [35]</td>
<td>SH</td>
<td>Intrinsic</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DPR [54]</td>
<td>SH</td>
<td>Intrinsic</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SIPR [43]</td>
<td>Environment Map</td>
<td>Im2Im</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Nestmeyer [29]</td>
<td>Directional Light</td>
<td>Intrinsic</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Hou [10]</td>
<td>SH</td>
<td>Im2Im</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Total Relighting [30]</td>
<td>Environment Map</td>
<td>Intrinsic</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Proposed</td>
<td>Directional Light</td>
<td>Intrinsic</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

A summary of our proposed method compared to recent face relighting methods.
where $x$ is a point on the 3D surface, $n$ is the surface normal at $x$, $\omega_i$ and $\omega_o$ are the incoming and outgoing lighting directions respectively, $\Omega$ is the unit hemisphere centered around $n$ with all possible values of $\omega_i$, $L_i(x, \omega_i)$ and $L_o(x, \omega_o)$ are the incoming and outgoing radiances respectively, and $f(x, \omega_i, \omega_o)$ is the bidirectional reflectance distribution function (BRDF) determining the material’s reflectance. If only diffuse reflection is considered, i.e., $f(x, \omega_i, \omega_o) = a(x)/\pi$, the rendering equation becomes:

$$L_o(x, \omega_o) = \frac{a(x)}{\pi} \int_{\omega_i \in \Omega} L_i(x, \omega_i) \cdot (n, \omega_i) \, d\omega_i = a(x)s(x).$$

(2)

Here $a(x)$ is the diffuse albedo and $s(x)$ the diffuse shading.

Assuming there is a dominant directional light in the scene, without considering the visibility of the light source from $x$, the diffuse shading $s(x)$ can be approximated by:

$$s(x) = i_a + i_d(n, \omega_d),$$

(3)

where $i_a$ is the intensity of the ambient light in the scene and $i_d$ and $\omega_d$ is the intensity and direction of the directional light, respectively. In other words, $L_i(x, \omega_i) = i_a + i_d \delta(\omega_i, \omega_d)$, where $\delta(\cdot, \cdot)$ is the Dirac delta function.

To model differentiable cast shadows, we introduce the shadow mask $M_{\text{shadow}}$ to model the visibility of the directional light. For each point, $M_{\text{shadow}}$ stores a value close to 0 if the point is under a cast shadow and close to 1 otherwise. The intensity under a cast shadow should be $i_a$, since the directional component is blocked by some part of the face and thus only the ambient component should contribute to the shadow’s intensity. To model cast shadows in the shading, we represent the modified shading $s'$ as:

$$s'(x) = i_a + M_{\text{shadow}}(x)i_d(n, \omega_d) = M_{\text{shadow}}(x)s(x) + (1 - M_{\text{shadow}}(x))i_a.$$  

(4)

Our reformulation of the rendering equation is thus,

$$L_o(x, \omega_o) = a(x)s'(x),$$

(5)

which overcomes the limitations of the diffuse shading $s$ and models cast shadows.

Unlike prior face relighting methods that model cast shadows [10, 29, 30], our formulation ensures geometrically consistent cast shadows by computing $M_{\text{shadow}}$ directly using the estimated face geometry. We discuss this in Sec. 3.3.3.

3.2. Architecture

Given a single image $I_t$ as input, our model estimates the intrinsic components: depth map $D_p$ (geometry), albedo $A_p$, the lighting direction $\omega_t$, and the ambient lighting intensity $i_{a,p}$. Our architecture is largely adopted from the hourglass network used by DPR [54], but we replicate the decoder and form two branches to estimate $D_p$ and $A_p$ respectively. $\omega_p$ and $i_{a,p}$ are estimated using a multilayer perceptron (MLP) following the encoder. The surface normals $N_p$ are then computed from $D_p$, and the shading $S_p$ is computed via Eqn. 4. We will discuss how $M_{\text{shadow}}$ is computed in the next section. The final rendered image $I_p$ is then generated following Eqn. 5 as:

$$I_p = A_pS_p.$$  

(6)

During training, we render $I_p$ using $\omega_p$, the estimated lighting direction of the input image $I_t$. We supervise the intrinsic component estimation by enforcing that $I_p$ reproduces the input image $I_t$. During inference, our model instead accepts a target lighting direction $\omega_t$ as input, which allows us to perform relighting. We compute $M_{\text{shadow}}$ using $D_p$ and $\omega_t$, which allows us to generate relit images with geometrically consistent cast shadows from any lighting direction. We illustrate our overall model architecture in Fig. 2.
3.3. Shading Estimation

One of our key contributions is producing geometrically consistent cast shadows using shadow mask \( M_{\text{shadow}} \). It is generated directly from the estimated geometry and used to produce shading \( S_p \) that models cast shadows. We now discuss the motivation and formulation.

Ray Tracing To motivate our algorithm for generating \( M_{\text{shadow}} \), we first discuss the traditional ray tracing algorithm. For every point \( x_i \) on the 3D object, the ray tracing algorithm casts a shadow ray towards the light source [1]. If the shadow ray intersects with a surface along its path, then \( x_i \) is under cast shadow. In our setting, the shadow rays determine which points are under cast shadow based on whether the ray intersects with some parts of the estimated 3D face, such as the nose.

Shadow Mask Estimation Our method incorporates the principles of ray tracing to generate \( M_{\text{shadow}} \) using the estimated depth map \( D_p \) and the target lighting direction \( \omega_t \) (see Fig. 3). Each pixel in \( D_p \) corresponds to a 3D point \( x_i \), and we represent \( \omega_t \) as a unit vector in 3D space. We can thus represent the shadow ray for point \( x_i \) as an order pair \((x_i, \omega_i)\). To determine whether it intersects the surface, we sample \( m = 160 \) points \( x_{i1}, x_{i2}, \ldots, x_{im} \) along its direction from \( D_p \) at regular intervals. We then provide a differentiable visibility function based on the following observation: if there exists a sampled point whose distance to the ray is close to 0, then the current ray or a nearby ray would intersect with the surface and the point \( x_i \) is under cast shadow. Conversely, if none of the sampled points have a distance close to 0 to the ray, the shadow ray does not intersect the surface and \( x_i \) is not under cast shadow. We therefore compute the minimum distance \( d_{\text{min}} \) between the sampled points and the ray by

\[
d_{\text{min}} = \min_{j \in [1, m]} |\overrightarrow{x_i x_j} \times \omega_t|,
\]

where \( \times \) is the cross product. If \( d_{\text{min}} \) is close to 0, we set the corresponding shadow mask value \( M_{\text{shadow}}(x_i) \) to be close to 0, indicating \( x_i \) is under a cast shadow. Otherwise, it should be close to 1. To achieve this while ensuring that computing \( M_{\text{shadow}}(x_i) \) is a differentiable operation, we define \( M_{\text{shadow}} \) as a Sigmoid function of \( d_{\text{min}} \):

\[
M_{\text{shadow}}(x_i) = \frac{-4e^{-d_{\text{min}}}}{(1 + e^{-d_{\text{min}}})^2} + 1.
\]

We apply our algorithm to all points \( x_i \) in depth map \( D_p \) to generate the shadow mask \( M_{\text{shadow}} \), which indicates where cast shadows lie on the face. Since we use \( D_p \) to compute \( M_{\text{shadow}} \), we directly leverage the 3D geometry of the face to synthesize our cast shadows, ensuring that they are geometrically consistent with respect to the face.

3.4. Training Losses

We utilize multiple loss functions to supervise the intrinsic decomposition. To supervise the depth estimation, we define \( \mathcal{L}_{\text{depth}} = \frac{\sum M_{\text{depth}} \|D_p - D_t\|}{\sum M_{\text{depth}}} \), where \( D_t \) is the groundtruth depth map, and mask \( M_{\text{depth}} \) defines where we have depth supervision in the image. The groundtruth depth is obtained using the method of Bai et al. [4] to first estimate the face mesh, and subsequently apply z-buffering.

To supervise the albedo estimation, we define \( \mathcal{L}_{\text{albedo}} = \frac{\sum M_{\text{face}} \|A_p - A_t\|}{\sum M_{\text{face}}} \), where \( A_t \) is the groundtruth albedo, and \( M_{\text{face}} \) is the full face mask. We generate our groundtruth albedo using SfSNet [35]. Since SfSNet’s estimated albedo does not generalize perfectly to our training data, we only apply \( \mathcal{L}_{\text{albedo}} \) in grayscale to give our model more freedom in estimating the RGB albedo.

To supervise our lighting estimation, we define two additional losses: \( \mathcal{L}_{\text{ambient}} \) and \( \mathcal{L}_{\text{light}} \). We define \( \mathcal{L}_{\text{ambient}} = \|i_{a,p} - i_{a,t}\|_1 \), where \( i_{a,t} \) is the groundtruth ambient intensity. Since determining the groundtruth ambient intensity of an image is challenging, we set \( i_{a,t} \) to be the same value for all training images. We also define \( \mathcal{L}_{\text{light}} = 1 - \langle \omega_p, \omega_t \rangle \), where \( \langle \omega_p, \omega_t \rangle \) is the inner product between the predicted and the groundtruth lighting direction \( \omega_t \). We obtain \( \omega_t \) using SfSNet, and convert the estimated SH representation into a dominant lighting direction.

To ensure that the estimated intrinsic components as a whole represent a plausible decomposition, we define a reconstruction loss \( \mathcal{L}_{\text{recon}} = \frac{\sum M_{\text{face}} \|I_p - I_t\|}{\sum M_{\text{face}}} \) between the rendered image \( I_p \) and the input \( I_t \). Finally, to improve the perceptual quality, we employ a PatchGAN [8] discriminator that operates on \( 70 \times 70 \) patches. We define our adversarial loss as \( \mathcal{L}_{\text{GAN}} \) and treat our rendered images as the
Nestmeyer et al. [29], two baselines modeling cast shadows. SIPR [43], DPR [54], and SfSNet [35] are unable to produce cast shadows. fake distribution and the input images as the real distribution. We also utilize a DSSIM loss to further improve the perceptual quality similar to Hou et al. [10] and Nestmeyer et al. [29] defined as \( \mathcal{L}_{DSSIM} = \frac{1}{2} (1-\text{SSIM}(\mathbf{I}_t, \hat{\mathbf{I}}_t)) \).

Our final loss function is thus defined as:

\[
\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{depth} + \lambda_2 \mathcal{L}_{albedo} + \lambda_3 \mathcal{L}_{ambient} + \lambda_4 \mathcal{L}_{light} + \lambda_5 \mathcal{L}_{recon} + \lambda_6 \mathcal{L}_{GAN} + \lambda_7 \mathcal{L}_{DSSIM},
\]

where \( \lambda_i \) are the weights for each loss function.

**Implementation Details** We train using PyTorch [31] for 100 epochs with the Adam Optimizer [15] and a learning rate of 0.0001 on one GeForce RTX 2080 Ti GPU. In training, we set \( \lambda_1 = 1, \lambda_2 = 5, \lambda_3 = 2.5, \lambda_4 = 1, \lambda_5 = 20, \lambda_6 = 0.01, \lambda_7 = 8 \), the groundtruth ambient intensity to \( i_{a,t} = 0.5 \), and the directional intensity to \( i_d = 0.5 \).

**4. Experiments**

Since our training objective is to minimize the difference between the rendered image \( \mathbf{I}_r \) and the input image \( \mathbf{I}_t \), we are ultimately free to use any face dataset with lighting variation. We train our model on the CelebA-HQ dataset [12], containing 30,000 in-the-wild face images from the CelebA dataset [25]. Following the testing protocol of Hou et al. [10], we evaluate our relighting performance quantitatively on the Multi-PIE [9] dataset, which contains 18 images per subject each with a unique directional light.

**Table 2. Relighting Evaluation on Multi-PIE Images with Target Lighting (mean ± standard deviation).** We compare our model against methods that accept a single image and a target lighting. Our method achieves the best performance across all metrics (bold).

**Figure 4. Qualitative Relighting Performance on Multi-PIE (Target Lightings).** Each method performs relighting given a single input image and a target lighting. Our method’s cast shadows much more closely match the target image compared to Hou et al. [10] and Nestmeyer et al. [29].

**Table 3. Lighting Transfer Evaluation on Multi-PIE (mean ± standard deviation).** Each input image is assigned a random reference image. The reference image is a different subject and a different lighting from the input image.
Shih et al. [38] transfers the wrong lightings. Furthermore, neither Shih et al. [38] nor Shu et al. [39] can produce cast shadows through lighting transfer, whereas our model can. Hou et al. [10] fails to transfer an appropriate cast shadow for the subject in the top row and the cast shadow for the subject in the second row is noticeably worse than ours in terms of shape and boundary.

Multi-PIE Evaluation Using Lighting Transfer Some methods require both an input and a reference image and relight by transferring the style of the reference to the input, known as lighting transfer. To evaluate our lighting transfer performance, we sample a random lighting for each Multi-PIE subject to serve as the input and a random reference image from the entire dataset. The reference image is a different subject from the input and under a different lighting. For lighting transfer, we first feed the reference image to our model to estimate the target lighting direction $\omega_t$ and the ambient intensity $i_{a,p}$. We then pass the input image along with $\omega_t$ and $i_{a,p}$ to our model to generate the relit image. The groundtruth target image is readily available in Multi-PIE. We compare with Shih et al. [38], Shu et al. [39], and Hou et al. [10] and report the results in Tab. 3. We achieve the best performance in MSE and comparable performance to Hou et al. in terms of DSSIM and LPIPS. We believe that a large reason why the performance is lower is the imperfect lighting supervision from SfSNet [35], which limits our model’s ability to estimate the correct lighting from the reference image and lowers the relighting performance.

4.2. Qualitative Evaluations

Multi-PIE Results We show qualitative relighting results of Multi-PIE [9] subjects for target lightings in Fig. 4 and for lighting transfer in Fig. 5. When using target lightings, our model produces cast shadows that much more closely match the shape of the shadows in target images compared to prior work. This is due to our shadow mask estimation that incorporates the face geometry to synthesize cast shadows, improving their geometric consistency. Hou et al. [10] and Nestmeyer et al. [29] also model cast shadows, but neither synthesize cast shadows directly from the face geometry and instead regress them from CNNs. Thus, they have no guarantee that their cast shadows correctly match the face geometry, as seen in Fig. 4. SIPR [43], DPR [54], and SfSNet [35] primarily model diffuse lightings and generally cannot produce cast shadows. When performing lighting transfer, we notice that our model can accurately estimate the target lighting from the reference image and is superior to the baselines in transferring over cast shadows (see Fig. 5). Shih et al. [38] and Shu et al. [39] largely fail to transfer over cast shadows and Hou et al. [10] produces cast shadows that do not match the shape of the groundtruth.

FFHQ Results We evaluate our performance on in-the-wild faces from the FFHQ [13] dataset. We show in Fig. 7 that we produce more geometrically consistent shadows than prior work across several subjects and lightings. We also show in Fig. 8 that our cast shadows are geometrically consistent as we rotate the target lightings around the face. Compared to Hou et al. [10], our cast shadows have much more plausible shapes and shadow boundaries.

4.3. Ablations and Additional Experiments

Reconstruction Error Analysis To better understand the distribution of reconstruction errors during relighting, Fig. 6
Figure 7. Qualitative Relighting Performance on FFHQ. Across multiple in-the-wild subjects and target lightings, our model produces more geometrically consistent cast shadows than prior methods while achieving noticeably better visual quality. Best viewed if enlarged.

Figure 8. Comparison of Geometric Consistency of Cast Shadows. We compare the geometric consistency of our cast shadows across 7 target lightings with Hou et al. [10], another face relighting method that models cast shadows. Notice that our model’s cast shadows shown in row b) are more plausible in terms of shape and shadow boundaries than the cast shadows of Hou et al., shown in row a).

visualizes the average $L_1$ error map between our relit images and the groundtruth target images in our Multi-PIE test set. We generate an error map for each target lighting separately and compute the average across all test subjects with that target lighting. We compare our error maps with Hou et al. [10], the SoTA face relighting method, and notice that our error maps have much lower error in the shadowed face regions, including the shadows cast around the nose. This further demonstrates that our method produces more geometrically consistent shadows across all test subjects.

**Geometry Error Analysis** One benefit of modeling hard shadows differentiably is that the end-to-end training may improve the intrinsic components, such as geometry, in face regions that cast hard shadows. To demonstrate this, we compare our surface normal errors on the Multi-PIE test images with two baselines: SfSNet [35], an intrinsic decomposition method with a diffuse SH lighting model, and DFNRMVS [4], which provides our geometry supervision. We choose surface normal error as the metric since the rendering equation uses surface normals, rather than the depth, to compute the shading. Although Multi-PIE lacks groundtruth 3D shapes, we use DFNRMVS [4] to estimate face meshes given 3 multi-view faces per subject as input, from which we compute the groundtruth surface normals. A dataset with large lighting variation and 3D groundtruth shapes is still lacking, partially due to the sensitivity of 3D scanners to illumination. We train on Multi-PIE subjects 1-250 and test on subjects 251-346. As for the geometry supervision in training, we use the face meshes from DFNRMVS provided only a single frontal image as input, which produces lower quality shapes than 3 views.

As shown in Tab. 4, across all test images and lightings, our model achieves the lowest average angular error in surface normal estimation. Improving over DFNRMVS shows that our model is not upper bounded by the quality of our shape supervision. Our end-to-end training incorporating differentiable shadow modeling can yield further improvements to the geometry. Improving over SfSNet also
Figure 9. **Per Lighting Surface Normal Error on Multi-PIE.** We compute the surface normal errors (degrees) across all test images of the same lighting. Our proposed model (P) achieves lower errors across all lightings than DFNRMVS [4] (D) and SiSNet [35] (S). We record the improvement percentage of P over the best baseline, where higher percentages are highlighted in brighter green. A reference image and its lighting direction are provided for each lighting. P improves the normals the most for lightings with hard shadows (e.g. lightings 4, 6, 8, 9, 10, 15, 17, and 18), which highlights the benefit of our hard shadow modeling in improving the face geometry.

<table>
<thead>
<tr>
<th>Method</th>
<th>Surface Normal Angular Error (Degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiSNet [35]</td>
<td>14.2796 ± 2.1442</td>
</tr>
<tr>
<td>Proposed</td>
<td>11.0672 ± 1.9489</td>
</tr>
</tbody>
</table>

Table 4. **Surface Normal Errors on Multi-PIE (mean ± standard deviation).** We compare with SiSNet and DFNRMVS. Our model produces more accurate surface normals than SiSNet, which assumes a diffuse SH lighting model, and DFNRMVS, our shape supervision, which shows the ability of our differentiable hard shadow modeling in improving the geometry.

Figure 10. **Surface Normal Improvement.** We visualize our surface normal improvement over the best baseline (DFNRMVS [4]) for test images in 4 lightings with hard shadows. The first row shows a reference image for each lighting. Notice the large improvements along and near the nose bridge and the face boundary, which cast hard shadows. This shows the contribution of our differentiable hard shadow modeling in improving the geometry.

5. **Conclusion**

We have proposed a novel face relighting method that produces geometrically consistent hard shadows. Unlike prior work, our approach is the first to directly synthesize cast shadows from the geometry, which improves the shadow’s shape and boundary. We have shown on the Multi-PIE and FFHQ datasets that our method achieves state-of-the-art face relighting performance quantitatively and qualitatively under directional lighting. We have also shown that our differentiable hard shadow modeling improves the geometry, especially around the nose, compared to prior work that assumes diffuse shading. We hope that our work will motivate future physics-driven relighting methods, and provide insights for handling hard shadows.

**Limitations** Since we use SiSNet’s [35] imperfect estimated lighting as supervision, our model accumulates more error during lighting transfer. The RGB albedo from SiSNet also does not generalize well to our training data, limiting the quality of our estimated albedo. Training on a dataset where we know the groundtruth lightings and could compute the albedo from photometric stereo similar to [29] would improve our model’s performance. In addition, although CelebA-HQ [12] contains some images under directional lighting, it primarily contains images under diffuse lighting. Our model would benefit from a publicly available in-the-wild dataset with primarily directional lights.

**Broader Impact** Creating deepfakes or affecting surveillance by adding shadows are major concerns. We acknowledge these risks but argue that our model only adds self shadows, which are generally limited in size and primarily around the nose. The user cannot freely manipulate the image or add shadows to any location they desire, which limits malicious use cases. Moreover, our method can synthesize images with self shadows for training, which can improve the robustness of face methods to self-shadowed images.
References

[10] Andrew Hou, Ze Zhang, Michel Sarkis, Ning Bi, Yiying Tong, and Xiaoming Liu. Towards high fidelity face relighting with realistic shadows. In CVPR, 2021. 1, 2, 3, 5, 6, 7
[21] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In ECCV, 2018. 2
[29] Thomas Nestmeyer, Jean-Francois Lalonde, Iain Matthews, and Andreas Lehrmann. Learning physics-guided face relighting under directional light. In CVPR, 2020. 1, 2, 3, 5, 6, 7, 8
[33] Laiyun Qing, Shiguang Shan, and Xilin Chen. Deep learning face attributes in the wild. In ICCV, 2015. 5, 6, 7, 8
[34] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David W. Jacobs. SISNet: Learning shape, reflectance and illuminance of faces in the wild. In CVPR, 2018. 1, 2, 4, 5, 6, 7, 8