Material Editing Using a Physically Based Rendering Network
#Supplemental File#

Guilin Liu[1,3*] Duygu Ceylan[2] Ersin Yumer[2] Jimei Yang[2] Jyh-Ming Lien[1]
[1]George Mason University [2]Adobe Research [3]NVIDIA

1. Details of Rendering Layer

We model the image formation process mathematically similar to the classical rendering equation, but without emitted spectral radiance. Given per-pixel surface normals $\mathbf{n}$ (in camera coordinates), material properties $\mathbf{m}$ and illumination $\mathbf{L}$, the outgoing light intensity for each pixel p in image I can be written as an integration over all incoming light directions ω_i:

$$I_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L}) = \int f(\overrightarrow{\omega}_i, \overrightarrow{\omega}_p, \mathbf{m}) \mathbf{L}(i) \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) d\omega_i, \tag{1}$$

where $\mathbf{L}(\omega_i)$ defines the intensity of the incoming light and $f(\omega_i, \omega_p, \mathbf{m})$ defines how this light is reflected along the outgoing light direction ω_p based on the material properties $\mathbf{m}$. ω_p is also the viewing direction, which can be computed using FOV and image size. In order to make this formulation differentiable, we substitute the integral with a sum over a discrete set of incoming light directions defined by the illumination $\mathbf{L}$:

$$I_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L}) = \sum_{\mathbf{L}} f(\overrightarrow{\omega}_i, \overrightarrow{\omega}_p, \mathbf{m}) \mathbf{L}(i) \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) d\omega_i. \tag{2}$$

where $d\omega_i$ represents the contribution (weight) a single light ω_i.

1.1. Representations

We now describe in detail how surface normals, illumination, and material properties are represented.

Surface normals (n). Given an image I of dimension $w \times h$, $\mathbf{n}$ is represented by a 3-channel $w \times h$ normal map where the r, g, b color of each pixel p encodes the x, y, and z dimensions of the per-pixel normal $\overrightarrow{n_p}$. The normal for each pixel has 3 channels:

$$\mathbf{n_p} = (n_p^1, n_p^2, n_p^3)$$

Illumination (L). We represent illumination with an HDR environment map of dimension 64×128. This environment map is a spherical panorama image flattened to the 2D image domain. Each pixel coordinate in this image can easily be mapped to spherical coordinates and thus corresponds to an incoming light direction ω_i in Equation 2. The pixel value stores the intensity of the light coming from this direction.

Let H_L and W_L represent the height and width of the environment map respectively. For each pixel $i = h * W_L + w$, which has the row index and column index to be h_L and w_L, in the environment map, we define θ_i^L and ϕ_i^L to be:

$$\theta_i^L = \frac{h_L}{H_L}\pi, \;\; \phi_i^L = \frac{w_L}{W_L}\pi$$

Then the direction of the lighting this pixel generates is:

$$\overrightarrow{\omega_i} =< \cos\phi_i^L \sin\theta_i^L, \cos\theta_i^L, \sin\phi_i^L \sin\theta_i^L >$$

Note that we will not compute the derivative of $\overrightarrow{\omega_i}$ and there is no parameter to learn during the training.

*This work was done when Guilin Liu was with Adobe and George Mason University.

Material ($\mathbf{m}$). We define $f(\omega_i, \omega_p, \mathbf{m})$ based on BRDFs [3] which provide a physically correct description of pointwise light reflection both for diffuse and specular surfaces. We adopt the Directional Statistics BRDF (DSBRDF) model [4] which is shown to accurately model a wide variety of measured BRDFs. The DSBRDF is based on approximating the BRDF values using the mixtures of hemi-sphere distributions.

To begin with, we define a *half vector* to be:

$$\overrightarrow{h_p} = \frac{\overrightarrow{\omega_i} + \overrightarrow{\omega_p}}{||\overrightarrow{\omega_i} + \overrightarrow{\omega_p}||}$$

We then denote the angle between half vector and lighting direction to be θ_d.

$$\theta_d = acos(min(1, max(0, \overrightarrow{\omega_i} \cdot \overrightarrow{h_p})))$$

The material coefficient is related to θ_d. For each θ_d, the material coefficient has 3 ($< R, G, B >$ channels) $\times$ 3 (3 mixtures of hemi-sphere distribution) $\times$ 2 (2 coefficients per hemi-sphere distribution) parameters. Instead of tabulating the material coefficients for every θ_d, we only estimate the coefficients of a few θ_d-s. Specifically, 18 θ_d-s' corresponding coefficients are estimated using the raw MERL BRDF dataset [2]. Later, those coefficients will be used to fit a second degree B-spline with nine knots, which results in 6 variables [1]. Thus, in total, there will be $3 \times 3 \times 2 \times 6$ parameters in each BRDFS's material coefficient $\mathbf{m}$.

We denote the second degree B-spline function as $S(\theta_d, \mathbf{m})$, then for any θ_d, the coefficients are $(m^k_{s,t}) = S(\theta_d, \mathbf{m})$, where $k \in \{0, 1, 2\}$, $s \in \{0, 1, 2\}$ and $t \in \{0, 1\}$. k represents one of the 3 channels (R, G, B); s represents one of the three mixtures of hemi-sphere distributions; t indexes two coefficients in hemi-sphere distribution.

The function $f(\overrightarrow{\omega}_i, \overrightarrow{\omega}_p, \mathbf{m})$ can be re-written as:

$$f^k(\overrightarrow{\omega}_i, \overrightarrow{\omega}_p, \mathbf{m}) = \sum_{s=0}^{2} (e^{m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}} - 1)$$

where k represents one of the 3 channels (R, G, B).

Having all of these representations, we can re-write our image formation as following:

$$I^k_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L}) = \sum_{i=1}^{H_L \cdot W_L} f^k(\overrightarrow{\omega}_i, \overrightarrow{\omega}_p, \mathbf{m}) \mathbf{L}(i) \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) d\omega_i$$

$$= \sum_{i=1}^{H_L \cdot W_L} (\sum_{s=0}^{2} (e^{m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}} - 1)) \cdot L^k(i) \cdot \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) \sin(\frac{\lfloor i/W_L \rfloor}{H_L} \pi)$$

1.2. Derivative

Derivative over Light. For lighting, we only need to compute the derivative over the intensity values of $L^k(i), k \in \{0, 1, 2\}$. We don't need to compute the derivative of the lighting direction.

$$\frac{\partial I_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L})}{\partial L^k(i)} = (\sum_{s=0}^{2} (e^{m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}} - 1)) \cdot \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) \sin(\frac{\lfloor i/W_L \rfloor}{H_L} \pi)$$

Derivative over Normal. We compute the derivative of normal for each channel individually. If $\overrightarrow{h_p} \cdot \mathbf{n_p} <= 0$ *or* $\mathbf{n_p} \cdot \overrightarrow{\omega}_i <= 0$, then $\frac{\partial I_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L})}{\partial n^c_p} = 0$

Otherwise,

$$\frac{\partial I_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L})}{\partial n^c_p} = \sum_{k=0}^{2} \sum_{i=1}^{H_L \cdot W_L} (\sum_{s=0}^{2} (e^{m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}} - 1)) \cdot L^k(i) \cdot \omega^c_i \sin(\frac{\lfloor i/W_L \rfloor}{H_L} \pi) +$$

$$\sum_{k=0}^{2} \sum_{i=1}^{H_L \cdot W_L} (\sum_{s=0}^{2} (e^{m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}} \cdot m^k_{s,0} \cdot m^k_{s,1} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1} - 1}) \cdot$$

$$L^k(i) \cdot \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) \sin(\frac{\lfloor i/W_L \rfloor}{H_L}\pi)$$

Derivative over Material. We first compute the derivative for $m^k_{s,t}$, and then based on chain rule and the spline interpolation function, we get the derivative for the original $\mathbf{m}$.

$$\frac{\partial I_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L})}{\partial m^k_{s,0}} = \sum_{i=1}^{H_L \cdot W_L} (e^{m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}) \cdot L^k(i) \cdot \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) \sin(\frac{\lfloor i/W_L \rfloor}{H_L}\pi)$$

$$\frac{\partial I_p(\mathbf{n_p}, \mathbf{m}, \mathbf{L})}{\partial m^k_{s,1}} = \sum_{i=1}^{H_L \cdot W_L} (e^{m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}}} \cdot m^k_{s,0} \cdot max(0, \overrightarrow{h_p} \cdot \mathbf{n_p})^{m^k_{s,1}} \cdot ln(\overrightarrow{h_p} \cdot \mathbf{n_p})) \cdot$$

$$L^k(i) \cdot \max(0, \mathbf{n_p} \cdot \overrightarrow{\omega}_i) \sin(\frac{\lfloor i/W_L \rfloor}{H_L}\pi)$$

then applying back the spline interpolation function $(m^k_{s,t}) = S(\theta_d, \mathbf{m})$, we have $\frac{\partial I_p(\mathbf{n_p},\mathbf{m},\mathbf{L})}{\partial \mathbf{m}} = (\frac{\partial I_p(\mathbf{n_p},\mathbf{m},\mathbf{L})}{\partial m^k_{s,t}}) \cdot \frac{\partial S(\theta_d,\mathbf{m})}{\partial \mathbf{m}}$

2. Network Design

We provide the detailed architectures for the normal, material, and illumination prediction modules in Figures 1, 2, and 3 respectively. Those networks can also be replaced with some other more recent network designs. The goal of this paper is to show performance improvement brought by the rendering layer.

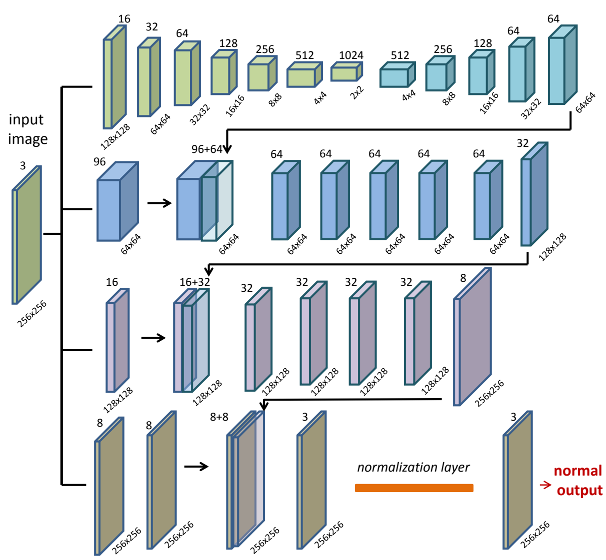

Figure 1. Architecture for the normal prediction module

3. Cross Material Transfer

In Fig 4, we provide the cross material transferring results, where we transfer the material predicted from one image to another image, using our approach with post-optimization.

References

[1] S. Lombardi and K. Nishino. Reflectance and natural illumination from a single image. In *IEEE ECCV*, pages 582–595. Springer, 2012. 2

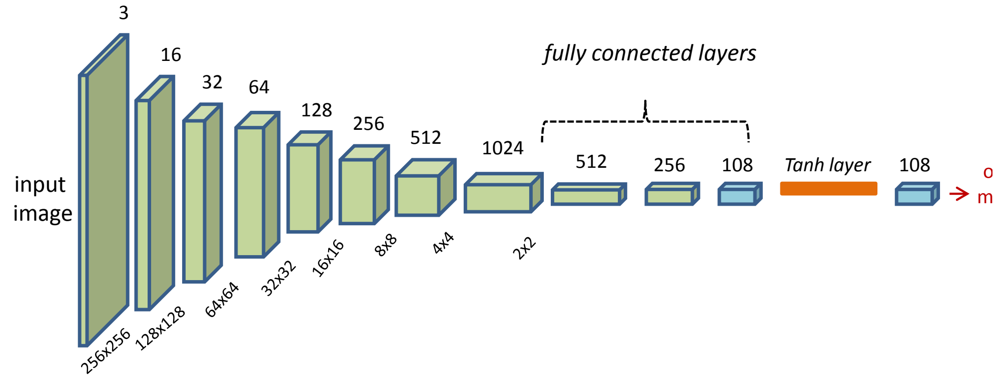

Figure 2. Architecture for the material prediction module

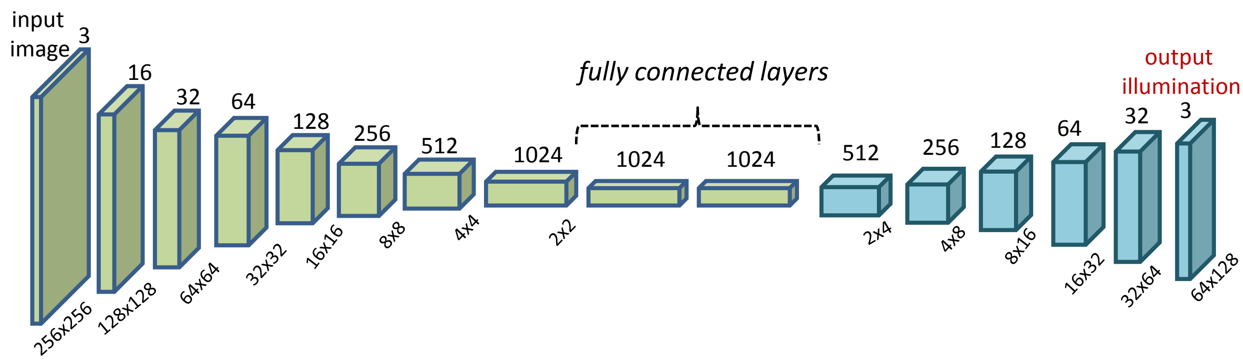

Figure 3. Architecture for the illumination prediction module

[2] W. Matusik, H. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003. 2

[3] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Radiometry. chapter Geometrical Considerations and Nomenclature for Reflectance, pages 94–145. Jones and Bartlett Publishers, Inc., USA, 1992. 2

[4] K. Nishino. Directional statistics brdf model. In *2009 IEEE 12th International Conference on Computer Vision*, pages 476–483. IEEE, 2009. 2

Figure 4. Given a set of images (in diagonal, in red boxes) as inputs, we synthesize new images by using shape and light from its row and material from its column using our approach.