

High-fidelity Face Tracking for AR/VR via Deep Lighting Adaptation

Lele Chen^{1,2} Chen Cao¹ Fernando De la Torre¹ Jason Saragih¹ Chenliang Xu² Yaser Sheikh¹
¹ Facebook Reality Labs ² Univeristy of Rochester

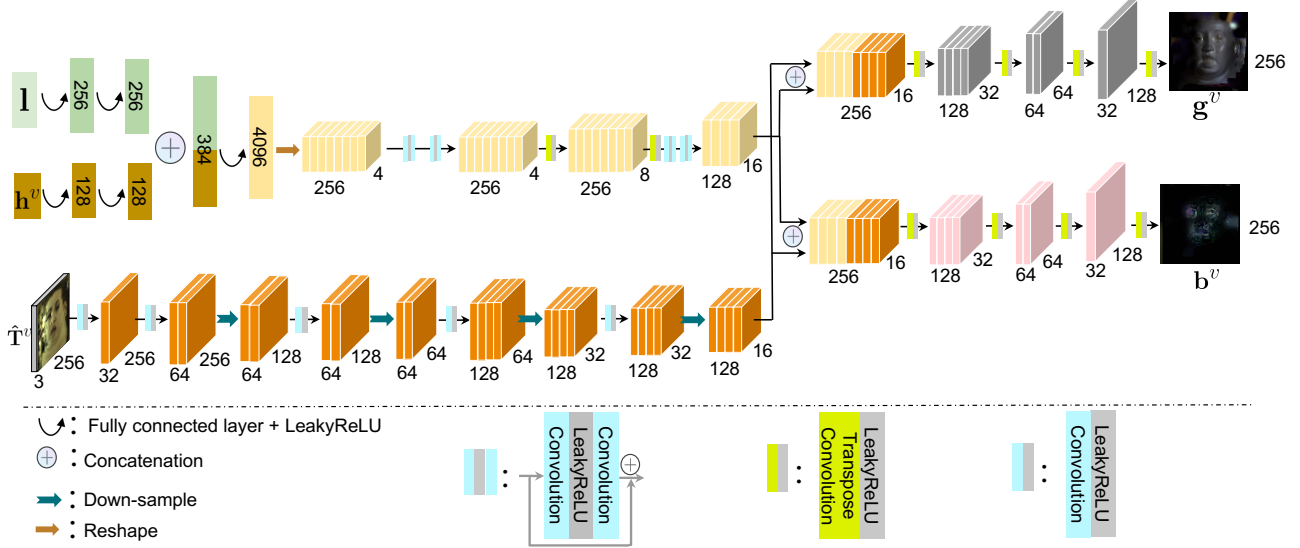


Figure 1. The detailed network structure of our lighting transfer network (G).

In this supplementary file, we make one typo correction in Sec. A, and explain the network details of our lighting model in Sec. B. Additionally, we show a demo video of our results in video_2987.mp4.

A. Corrigenda

The Eq. 7 in the submission should be:

$$\mathcal{L}_{\text{pix}}(\mathbf{l}, \phi, \{\mathbf{p}_k\}) = \sum_k \|r_k \odot w_k\|_1 + \lambda_{\Delta} \|\Delta r_k \odot w_k\|_1, \quad (1)$$

where $r_k = I_k - \hat{I}(\mathbf{p}_k)$ is the reconstruction residual, w_k is the foreground mask, and $\mathbf{p}_k = [\mathbf{r}_k, \mathbf{t}_k, \mathbf{z}_k]$.

B. Lighting model G

B.1. Network Structure

We present the detailed network structure in Fig. 1.

B.2. Inputs and Outputs

The lighting code \mathbf{l} is a pre-defined vector when we train G on light-stage data, and is a learnable vector when

we refine G on in-the-wild video frames. During training on light-stage data, the lighting direction is encoded by the position of the non-zero element in \mathbf{l} , and the lighting color is encoded by the value of the non-zero element in \mathbf{l} . The view-dependent head pose $\mathbf{h}^v \in \mathbb{R}^6 = \{\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z, \mathbf{v}_x^v, \mathbf{v}_y^v, \mathbf{v}_z^v\}$, where $\mathbf{r} = \{\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z\}$ and $\mathbf{v}^v = \{\mathbf{v}_x^v, \mathbf{v}_y^v, \mathbf{v}_z^v\}$ are rigid head rotation and viewpoint vector, respectively. The fully-lit texture $\hat{\mathbf{T}}^v$ is obtained from DAM decoder, and we down-sample it to the size of $3 \times 256 \times 256$.

The outputs are the gain and bias map $\mathbf{g}^v, \mathbf{b}^v$, and we upsample the output $\mathbf{g}^v, \mathbf{b}^v$ back to the size of $3 \times 1024 \times 1024$ by bilinear interpolation.