HDR Environment Map Estimation for Real-Time Augmented Reality Supplementary Appendix

Gowri Somanath
Apple
gowri@apple.com

Daniel Kurz
Apple
daniel_kurz@apple.com

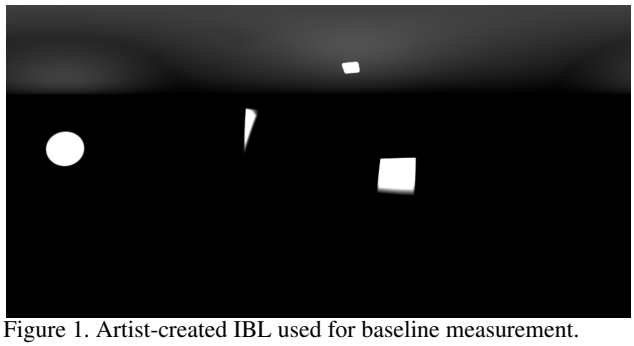

Figure 1. Artist-created IBL used for baseline measurement.

1. Video of mobile application

A video providing an overview of our method and a demonstration of our mobile application in the real world using iPhone XS is available at https://docs-assets.developer.apple.com/ml-research/papers/hdr-environment-map.mp4

2. Artist-created environment map

In Figure 1 we show the artist-created environment map used as a baseline in our benchmarking. The artist designed it to satisfy aesthetic and lighting requirements. For lighting, the intensities are selected to make sure the objects were well exposed and that middle gray is retained. The aesthetic brief was to have a "studio lighting" feel with a broad area light from behind the camera and from above.

EnvMapNet-conv-block
short-cut,x=input Repeat 5 times: x=BatchNormalization(x) x=LeakyReLu(slope=0.2)(x) x=Convolution(kernel=(3,3),filters=16)(x) x=Concatenate(x,short-cut) short-cut=x output=x
EnvMapNet-downsample-block
x=input x=Convolution(kernel=(3,3),filters=dk)(x) x=AveragePool2D(x) output=x
EnvMapNet-upsample-block
x=input x=NearestNeighbourUpsample2x(x) x=Convolution(kernel=(3,3),filters=uk)(x) output=x
Discriminator-residual-block
sc=AveragePool2D(input) sc=Convolution(kernel=(3,3),filters=ak)(sc) x=input Repeat 2 times: x=BatchNormalization(x) x=LeakyReLu(slope=0.2)(x) x=Convolution(kernel=(3,3),filters=ak)(x) output=Add(x,sc)

Table 1. Building blocks used in EnvMapNet and discriminator.

3. Network details

In Table 1 we provide the building blocks of the model architectures used in our method. Spectral Normalization [4] is used in all the convolutional layers.

Our proposed model, EnvMapNet, consists of an encoder and decoder as shown in paper Figure 2. The encoder is composed of five sets of *EnvMapNet-conv-block* and *EnvMapNet-downsample-block*, with uk=[64, 128, 128, 128, 256, 256, 512] for each consecutive block respectively. The resulting latent vector is convolved with a 1×1 kernel to output 64 filters. The decoder mirrors the encoder by using *EnvMapNet-conv-block* and *EnvMapNet-upsample-block*. The final output is produced by a 3×3 convolution to produce 3 channels for RGB, followed by a tanh activation. We use skip connections between same sized layers of the encoder and decoder.

The discriminator is composed of residual blocks with ak=[64, 128, 256, 256, 256, 256, 256] for consecutive blocks respectively. The outputs from the discriminator are

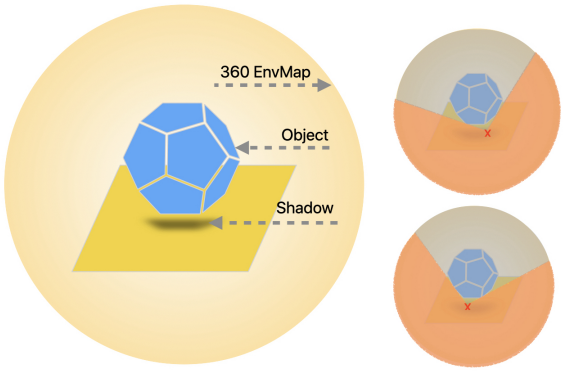

Figure 2. Intuitive understanding of ProjectionLoss and its relation to shadow casting. The loss is defined in Section 3.3.

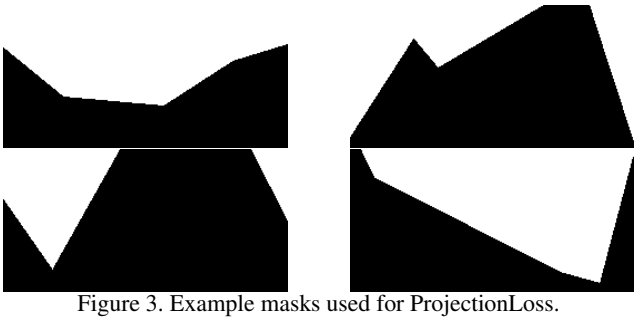

Figure 3. Example masks used for ProjectionLoss.

the binary classification of real or fake, and the classification into the K-means cluster ID. Each is obtained by convolution layer with the corresponding number of output channels and global average pooling.

4. ProjectionLoss and user study details

In this section we expand on the background for our proposed ProjectionLoss, its relation to shadow casting, and the details of the user study and experiments discussed in paper Section 3.3.

4.1. Relation to shadows

In Figure 2 we provide an illustration to explain our motivation for ProjectionLoss. Consider an object (blue ball), lit by an environment map, and casting a shadow on the planar surface below. Accurate generation of shadows requires for every point on the plane (red cross) the computation of the integral over the part of the env map that is visible from that point (shaded orange), i.e. not occluded by the object (blue ball). To this end, the environment map is element-wise multiplied with a visibility mask. Our precomputed randomized masks (Figure 3) used in projection loss are examples for such visibility masks for different points on the plane and different object shapes. As a result, ProjectionLoss encourages our predicted env maps to lead to shadows similar to ground truth.

4.2. User study

To further understand the value of ProjectionLoss for lighting estimation, and to compare to other measures, such as SSIM [8] and Mean Squared Error (MSE), we performed the following experiments. For each dataset environment map, we render an image of a scene with multiple geometric objects as shown in Figure 4.

First we establish a baseline metric to measure similarity between rendered images. We propose using the SSIM score, which is a metric often used to measure human perception and image similarity, to quantify the similarity of our rendered images. To validate this for our application, we perform a user study asking participants to choose from four provided options which rendered image looks most similar to a reference rendered image. The four options were randomly selected such that two of them were in the top-10 as retrieved by SSIM, and the other two from outside the top-10. Based on the results of 5 study participants, each providing their selection for 380 reference images, we found that on average human participants selected one of the top-10 SSIM images 95% of the time. Hence retrieval of similar environment maps based on SSIM between rendered images is a baseline method to find environment maps that produce similar lighting on the objects.

We then evaluate the correlation between SSIM on rendered images with ProjectionLoss and MSE calculated on the corresponding (equirectangular) environment maps. We can observe that the retrieval of similar environment maps by ProjectionLoss is better matched with retrieval based on SSIM compared to using MSE, see Figure 4. Quantitatively, we found the intersection of top-5 retrievals by SSIM (on the rendered images) and those using ProjectionLoss (on the environment map) to be 1.6 ± 0.7, while it was 0.6 ± 0.5 using MSE (on the environment map).

Based on the above we believe that our proposed ProjectionLoss is a good loss to train the network for estimating lighting such that the end result for rendering is accurate with respect to ground truth. Secondly, we show that MSE on the environment map is insufficient for training accurate light estimation, and its use as a metric of comparison or benchmarking, as done in previous works, would not correlate well with the final application.

5. Comparisons with recent methods

In this section we expand on the qualitative results from paper Section 5.1 and provide comparisons to the very recent work of Srinivasan *et al*. [6], that trained and evaluated on synthetic LDR images from InteriorNet [3]. The authors also provided results from their re-implementation of Deep Light [2] and Neural Illumination [5]. We used the images and results provided as part of their paper and show the comparison to our results in Figure 5. We note that the

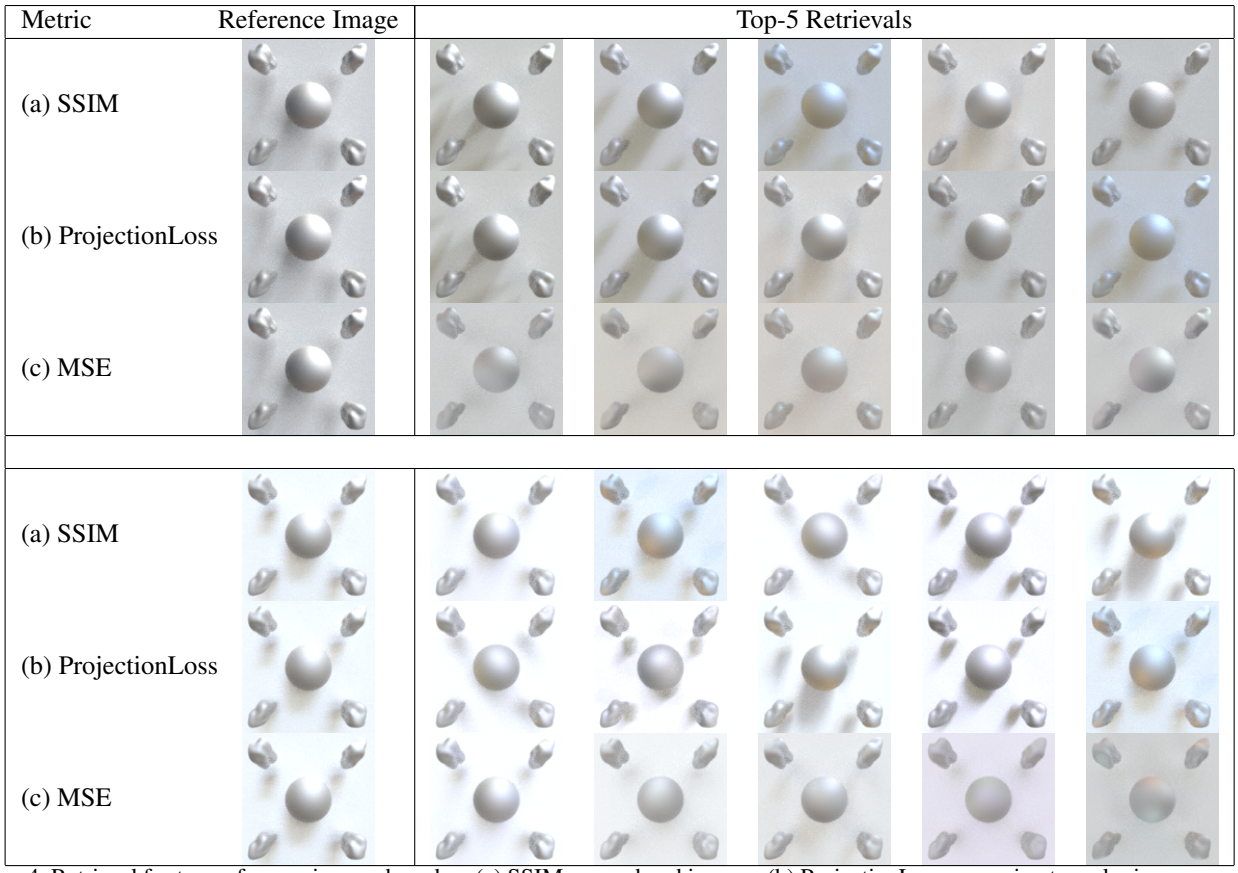

Figure 4. Retrieval for two reference images based on (a) SSIM on rendered images, (b) ProjectionLoss on equirectangular images, and (c) MSE on equirectangular images. A user study on 380 reference images showed 95% agreement with SSIM retrieval. Average intersection of participants' selection of most similar rendered images with top-5 retrieval from ProjectionLoss was 1.6, and from MSE was 0.6.

input to our algorithm was only the incomplete panorama shown in Figure 5(a), while stereo images were used for Srinivasan *et al*. [6] (c), and the 32×32 sphere image output from LeGendre *et al*. [2] was converted to an equirectangular projection by the authors (d). Furthermore, compared methods were (re)trained by Srinivasan *et al*. on the same synthetic dataset (InteriorNet [3]), while our results are using the same network as before. It was trained on $2,810$ images from public datasets of real-world images - Laval Indoor HDR dataset [1] and PanoContext LDR panoramas [7] - as discussed in paper Section 3.1.

Since the network used by Srinivasan *et al*. was trained on stereo input and LDR ground truth (with reverse gamma), we only make qualitative comparisons for LDR completion/generation of environment maps. We noticed that the predictions made by the algorithm from Srinivasan *et al*. resembles the (unseen) ground truth textures very closely. For example, the window, curtains and wall boundaries in the first two images shown in Figure 5(c) closely match the corresponding regions in Figure 5(b). This is surprising given that the input panorama does not contain this information.

Note that our model was trained with images which are aligned with vertical axis being gravity, that is, the horizon is aligned to the horizontal image axis. This is easily achieved in mobile AR using device pose and orientation, and it also avoids ambiguity when training a network. The images provided as input (a) have different rotations resulting in some of the artifacts observed. Additionally we observe that the input panoramas are not just masked subset of pixels from the ground truth. The artifacts in the input, such as the blocks and aliasing seen on the left side of the first input, could potentially be from reprojection done by Srinivasan *et al*., which is carried forward by our results as part of 'known input'. We manually corrected the above by 'straightening' the panoramas (g), and using the input mask to create Corrected Input (h), and show results from our method using this corrected input in row (i). Qualitatively we can observe that our method can produce plausible and perceptually pleasing reflections in these new scenes, even though our model was not trained on the same dataset.

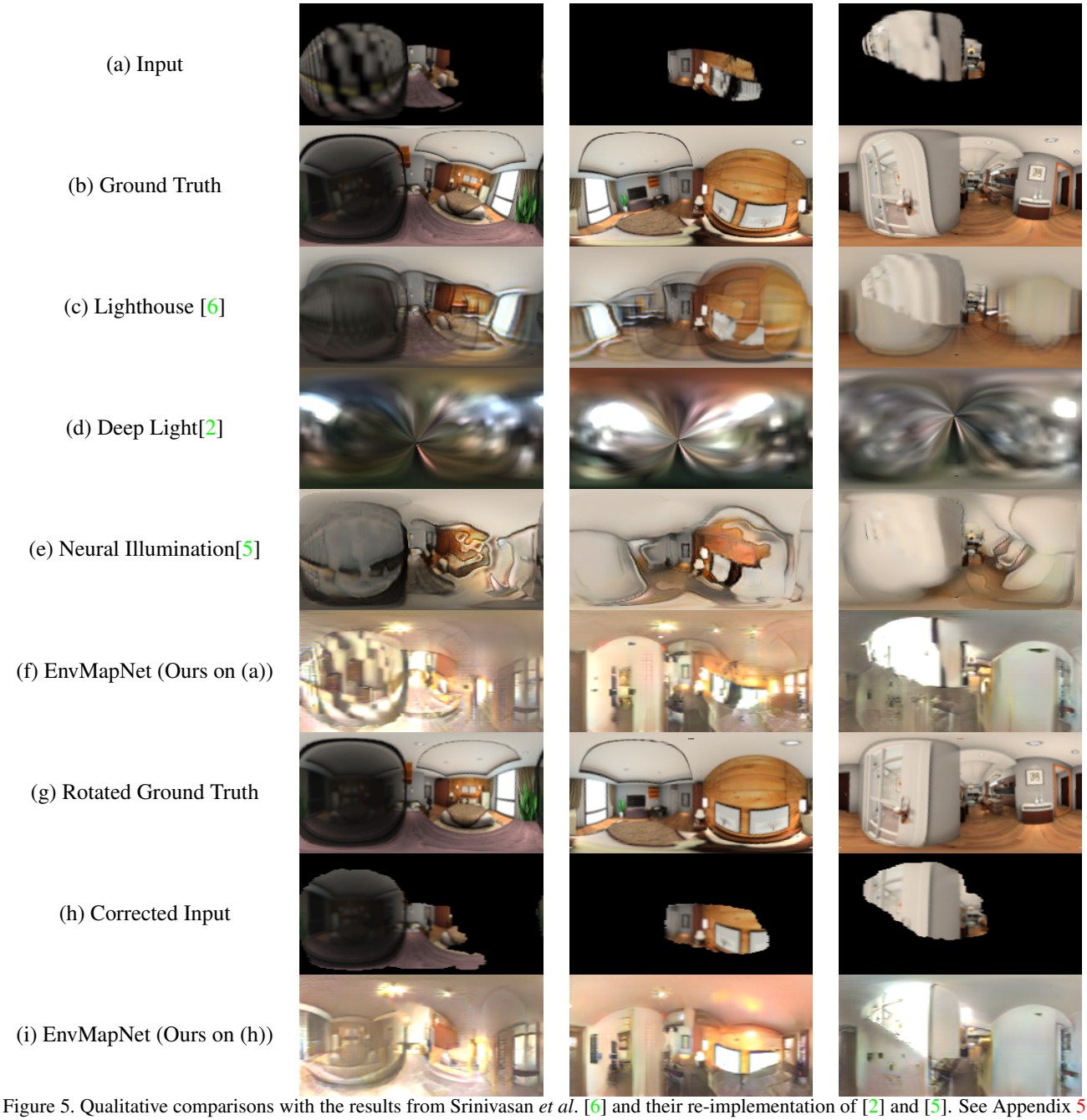

Figure 5. Qualitative comparisons with the results from Srinivasan *et al*. [6] and their re-implementation of [2] and [5]. See Appendix 5.

6. Results and comparison

In this section we provide several images to demonstrate the effectiveness of our method, for rendering virtual objects in mobile AR applications, and expand on the images and results discussed in paper Section 5.2.

In Figures 6 and 7 we show predicted environment maps from Gardner *et al*. [1] and our method over a variety of scenes and lighting conditions. We further show their use for rendering both reflective (teapot) and diffuse (dragon) objects[1] with shadows. Each result is shown over a pair of rows containing the input crop, rendered images, and corresponding environment maps with angular error below each. To enable fair and future comparison, we crop the test environment maps in the center for 90 degree FoV. We rectify and provide the **Input Crop** to [1], and project them to a equirectangular map for our method. We obtain the predicted equirectangular maps from each image, extract

[1] http://graphics.stanford.edu/data/3Dscanrep/

the parametric lights for calculation of AngularError as described in paper Section 4, and proceed to use for rendering.

We use Tungsten[2] an open source path tracer, to generate renderings of an aluminum metal finish teapot and a lambertian material on the dragon objects. Only for the purpose of rendering, we perform the following post-processing operations on each predicted equirectangular map. Since each method provides the result in an arbitrary intensity range, we scale the pixel intensities for each predicted result such that the average intensity in the region provided as input is equal to that of the ground truth HDR. We further overlay the pixels from the ground truth corresponding to the input FoV, to simulate the AR video backdrop and common background for each rendering. We provide this equirectangular map as the input for Tungsten, position the virtual camera to match the input crop provided, and obtain the rendered images shown in Figures 6 and 7 for **EnvMapNet (Ours), Ground Truth** and **Gardner *et al.* [1]**. We stress that this post-processing was only done for the offline rendering pipeline and not for other comparisons.

As previously detailed, this simulates that a virtual object is placed at the center of a probe which is illuminated using the environment map. As shown in the images, our results generate perceptually pleasing and accurate lighting for the virtual object compared to those from [1], for both material finishes. Though our method cannot (and should not) create the unseen ground truth scene exactly, the level of detail generated is clearly plausible and provides a better visual coherence compared to those from [1]. This aspect regarding image resolution and quality of the details is captured by the FID metric as detailed in the main paper. The renderings of the diffuse dragon object on a plane highlight the difference in accuracy of estimated light direction, as captured by our AngularError metric listed below each result. The images show estimated environment maps with a wide range of errors from both methods, and as can be observed, overall our estimates produce shadows which better match those from the ground truth environment map.

References

[1] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2017. 3, 4, 5, 6, 7

[2] Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul Debevec. Deeplight: Learning illumination for unconstrained mobile mixed reality. In *ACM Transactions on Graphics*, 2019. 2, 3, 4

[3] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *Proc. BMVC*, 2018. 2, 3

[4] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. ICLR*, 2018. 1

[5] Shuran Song and Thomas Funkhouser. Neural illumination: Lighting prediction for indoor environments. *Proc. CVPR*, 2019. 2, 4

[6] Pratul P. Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T. Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *Proc. CVPR*, 2020. 2, 3, 4

[7] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Proc. ECCV*, 2014. 3

[8] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 2

[2] https://github.com/tunabrain/tungsten

Figure 6. Results with variety of angular errors. Each result is shown over a row containing the input crop, rendered images with reflective teapot and diffuse dragon, with corresponding environment maps below each. As discussed in paper Section 5 our method is qualitatively and quantitatively able to produce visually coherent environment maps for lighting, shadows and reflection.

Figure 7. More results with variety of angular errors. Each result is shown over a row containing the input crop, rendered images with reflective teapot and diffuse dragon, with corresponding environment maps below each. See Figure 6 and paper Section 5.