HDR Environment Map Estimation for Real-Time Augmented Reality

Gowri Somanath
Apple
gowri@apple.com

Daniel Kurz
Apple
daniel_kurz@apple.com

Abstract

We present a method to estimate an HDR environment map from a narrow field-of-view LDR camera image in real-time. This enables perceptually appealing reflections and shading on virtual objects of any material finish, from mirror to diffuse, rendered into a real environment using augmented reality. Our method is based on our efficient convolutional neural network, EnvMapNet, trained end-to-end with two novel losses, ProjectionLoss for the generated image, and ClusterLoss for adversarial training. Through qualitative and quantitative comparison to state-of-the-art methods, we demonstrate that our algorithm reduces the directional error of estimated light sources by more than 50%*, and achieves* 3.7 *times lower Frechet Inception Distance (FID). We further showcase a mobile application that is able to run our neural network model in under* 9 *ms on an iPhone XS, and render in real-time, visually coherent virtual objects in previously unseen real-world environments.*

1. Introduction

In this work, we discuss video see-through augmented reality (AR) applications, in which virtual objects are superimposed on camera frames of the real environment shown on an opaque display, e.g. on a phone as shown in Fig. 1(e). Creating immersive and believable AR experiences involves many aspects of computer vision and graphics. One of the requirements is visual coherence: the problem of matching visual appearance of rendered objects to their real-world background, such that virtual and real objects become indistinguishable in the composited video. Accomplishing this involves matching various scene and camera properties, such as lighting, geometry, and sensor noise.

This paper focusses on creating reflections and lighting for virtual objects by estimating an omnidirectional HDR environment map. To support rendering objects with a variety of geometry, material properties, and dimensions, the environment map must be high dynamic range, and have sufficient image resolution to represent objects and features in the scene. We use the equirectangular projection and RGB color space for the environment maps. As shown in Fig. 1(a), the challenge in mobile AR is limited camera field of view (FoV) and motion by the user, hence an application is usually able to accumulate less than 100 degrees effective FoV. A virtual object placed in front of the user, however, is expected to reflect what is behind the camera, and parts which are not present in the captured frames. The problem is thus to estimate, given this incomplete environment map, a plausible estimation for rest of the scene and its lighting. We show that our method is not only able to estimate the light information, but also to synthesize a high resolution completed scene. For instance in the scene shown in Fig. 1(b), the estimated environment map is high resolution, continuous, and a plausible extrapolation of the input. The synthesized parts not only match low frequency information (ambient light temperature and intensity), but also finer details such as the type of light sources (in this case, ceiling area lights). As detailed in Sec. 3, we achieve this context-aware scene completion using the framework of generative adversarial networks (GANs) [12] along with novel loss functions, ProjectionLoss and ClusterLoss, designed for accurate light estimation.

In mobile AR frameworks [1, 2], we can obtain camera frames, poses and scene geometry, around the 3D location where the virtual object is to be placed. This allows a real-time renderer to create light probes [27] at the 3D location. RGB texture information from the frames can be rendered into an equirectangular image at these probe locations. In this work, we focus on processing the partial environment map at these probes. Our method takes as input a partial environment map that is composed from one or more low dynamic range (LDR) camera frames (8 bit per channel), and outputs a completed environment map that is higher dynamic range (HDR, 16 bit channel). Thus we perform both lifting of input pixels from LDR to HDR, as well as spatial HDR image extrapolation. The output environment map retains the color and details from pixels that were in the input, while filling the unknown pixels with plausible content that is coherent with the known. That is, we want the completed environment map to represent the textures from a plausible real scene. Through detailed quantitative and

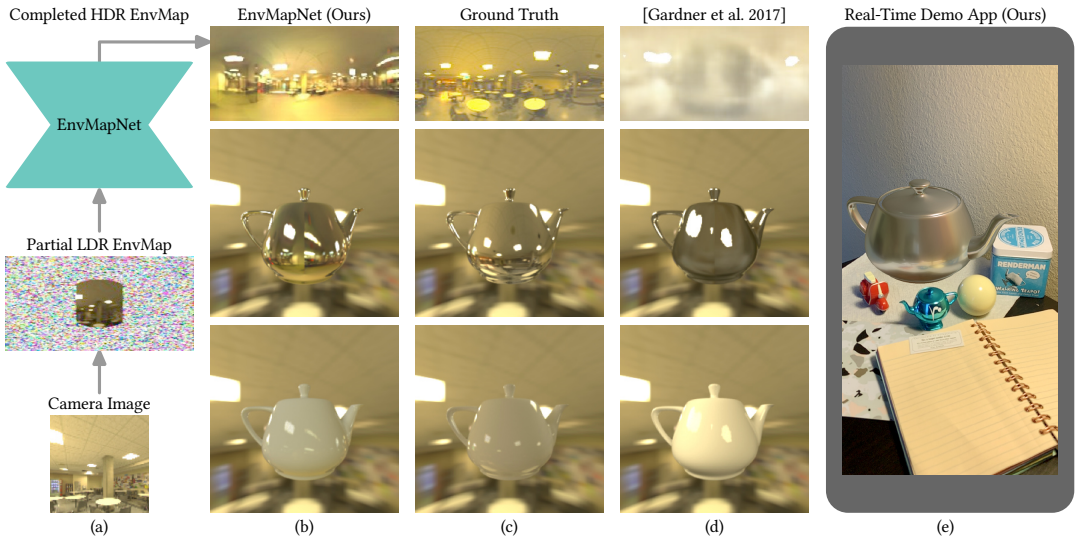

Figure 1. Given a partial LDR environment map from a camera image (a), we estimate a visually coherent and completed HDR map that can be used by graphics engines to create light probes. These enable rendering virtual objects with any material finish into the real environment in a perceptually pleasing and realistic way (b), more similar to ground truth (c) than state-of-the-art methods (d). Our mobile app renders a virtual teapot using our estimated environment map in real-time (e). **See supplementary material for videos.**

qualitative comparisons, we demonstrate that our method surpasses the current state-of-the art to estimate high quality, perceptually plausible, and accurate HDR environment maps. We reduce the directional (angular) error for lights by more than half, and achieve a significantly lower Frechet Inception Distance (FID).

Related works estimate a subset of the properties, such as lighting direction, color, and intensity using low resolution images [11, 9, 17], parametric lights [15, 9], or LDR environment maps [25]. Methods that estimate both light and texture [10, 17, 23] lack resolution and detail in the estimated environment map. In contrast, our solution provides sufficient details in the image for virtual objects of any material finish including reflective mirrors. To create plausible scene completions, we take inspiration from work in the area GANs for image synthesis. However, most success for GANs has come for datasets with single objects (e.g. faces [16]), using conditional labels [21], or self-supervision [5]. Its direct use for panoramic images of indoor scenes, small datasets with high appearance variation, and ambiguity in semantic labeling, is challenging. We present ClusterLoss, a novel training loss for the discriminator, that allows us to create realistic images with a dataset of only $\sim$2,800 training images.

We also observe that each of the previous methods have a different evaluation scheme, are based on respective representations and/or involve subjective user studies. Though subjective plausibility is important for user experience, it makes benchmarking difficult. We present metrics that can be used to quantitatively compare both the lighting and reflection quality of environment maps.

In summary, we make the following core contributions:

- We present a method to generate an HDR environment map suitable for both reflections and lighting from a small FoV LDR image or partial environment map. To our knowledge, ours is the first to achieve this for real-time AR applications (under 9ms on iPhone XS).
- We present two novel contributions to the training pipeline in the form of ProjectionLoss for the environment map, and ClusterLoss in the adversarial training. This allows us to reduce the directional error by more than 50% compared to current methods, while achieving 3.7 times lower FID.
- We establish metrics that allow easy quantitative comparisons with related work, and thus provide a systematic benchmark for this emerging area in AR.

2. Related work

A classical technique to obtain an HDR environment map is to merge images of a mirror sphere in the scene captured under multiple exposure brackets [7]. This can be applied for offline use cases but is unsuitable for real-time mobile AR in arbitrary and novel environments. Some previous works have used additional cues about an object [20, 26], scene geometry [3, 19, 30], or special cases such as sun position estimation [31, 15, 14]. For brevity, we only discuss works that focus on light estimation from small FoV images of general scenes. In this context, works can be divided into two categories: those that focus on light estimation with low dimensional parameters, and those which estimate both lighting and environment maps.

Garon *et al*. [11] use spherical harmonics representation for light and depth estimated using the SUNCG dataset [24][1]. Cheng *et al*. [6] also predict 48 spherical harmonics coefficients from two images captured by the front and rear cameras of a mobile device. Gardner *et al*. use parametric lights as representation [9]. The parameters are derived using peak finding, region growing, and ellipse fitting on intensity images. Even though use of parametric lights reduces the decoder size, the authors use L2 loss by converting the parameters to an equirectangular image. It is not clear how to extend this method to generate higher quality texture in the environment map that would be consistent with the regressed parameters. We thus choose to have an end-to-end network that directly estimates the HDR environment map. We use the same parametric lights to create quantitative metrics for benchmarking.

Gardner *et al*. use equirectangular representation [10] by dividing the task into light position estimation (trained with LDR panoramas), and HDR intensity estimation (trained with Laval HDR Images dataset). This is the work most similar to our method in input-output, representation, and formulation. In contrast to their method, we have a single stage training from an LDR small FoV image to a completed HDR environment map, and using our proposed adversarial training we are able to generate more realistic RGB scene completion for use on reflective virtual objects. Some recent works, like ours, employ an adversarial loss to generate a completed environment map [23, 25] or sphere images [17]. LeGendre *et al*. captured a special dataset with three spheres of mirror, matte, and diffuse gray finish [17]. They train a network to regress from a camera image to three small (32×32 pixel) images representing the sphere segments. This precludes use of the results for high quality reflections. Also, due to the low resolution, the estimate covers mostly the portion of the scene behind the camera. It is useful to render a given frame, but as the camera commonly moves with respect to the virtual object in a AR application, a new estimate has to be inferred frequently. Thus even with the assumption of a static scene, their method would suffer from temporal flickering due to multiple independent estimates.

Song *et al*. use a multi-stage ensemble that estimates geometry, LDR completion and HDR illumination [23]. Our method uses a single model, and we do not require depth map per HDR image for training. This makes our method efficient to train and use in real-time mobile AR. The authors do not specify the resolution of the panoramic images, but visual observation shows lack of resolution and artifacts due to the projection from noisy 3D reconstructions. Srinivasan *et al*. use an input stereo pair to generate the output environment map [25]. They use the LDR images from the synthetically generated InteriorNet dataset [18] to train the light estimation, by application of inverse gamma on the tone mapped images. This limits the ability to learn realistic high dynamic range and accurate lighting, as also indicated by their results on mostly specular objects. In contrast, our method does not require stereo input, and estimates an HDR environment map that can be applied for lighting objects with wide range of materials from diffuse to mirror.

Furthermore, we employ only 2,100 HDR images of the publicly available Laval HDR Images dataset of real environments, a much smaller dataset than the 4.06 million non-public specialized sphere images used in [17], the withdrawn SUNCG used in [24], or the 200,000+ Matterport images in [23]. The lack of shared code, model weights and use of private datasets make it difficult to make comparisons to these recent methods, that also compare to [10] as we do in our quantitative evaluations. However, we present qualitative comparisons to these works in Sec. 5.

With the exception of [17], that outputs a 32×32 low resolution sphere image, no other method has been demonstrated on mobile devices. To the best of our knowledge, we are the first to demonstrate real-time on-device generation of environment maps of high resolution and image quality.

3. Proposed method

Creating a light probe in a mobile AR application involves two broad stages: first to select a 3D point as center of this probe and project known scene information to an equirectangular environment map with the selected point as camera center; and second to process the partial equirectangular to output a completed HDR environment map. The first requires color and scene geometry knowledge or assumption for projection. In our work, we assume without loss of generality that platform and application-dependent processing can be used for this projection, and obtain the incomplete environment map from the probe center. Using the mobile device pose also helps to ensure that the environment maps are upright or gravity aligned. That is, the floor always appears at the bottom and the ceiling on top. Our focus in this paper is on completion, and LDR to HDR lifting of this incomplete environment map as shown in Fig. 2.

We use the equirectangular representation (128×256 pixels) and create a four channel input (RGB-mask). Known pixel intensities are normalized to $[-1.0, 1.0]$, while the unknown pixels are populated with random noise from a uniform distribution $U(-1.0, 1.0)$. The fourth channel is a binary mask with known pixels set to 0. This is input into our network, EnvMapNet, that outputs an HDR RGB image in log scale. The log image is converted to linear values, optionally decomposed into analytical lights, and provided to the renderer. We train our network end-to-end with image-based and adversarial losses, which allows us to handle both the generative aspect for reflection completion as well as light estimation from partial environment map.

[1] SUNCG is currently withdrawn from distribution

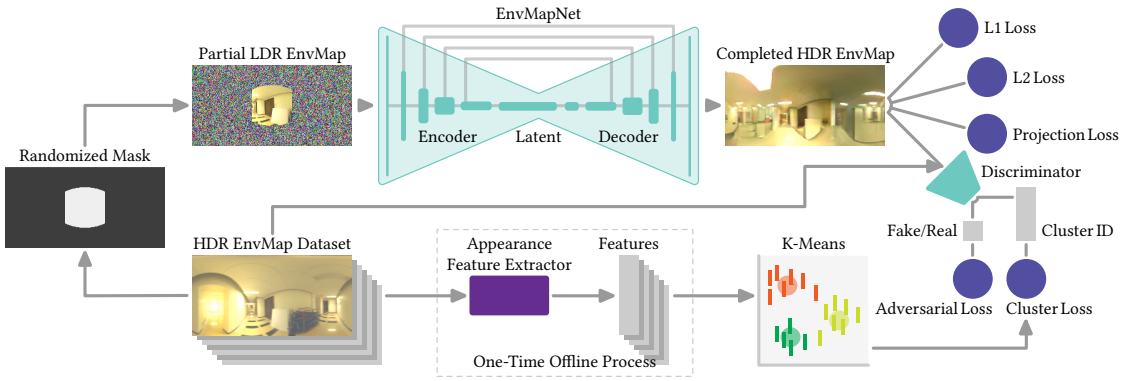

Figure 2. Overview of our method: We propose EnvMapNet that estimates completed HDR environment maps from partial input. We train the network end-to-end in an adversarial setup. An offline one-time clustering of the training images is used to provide a supervised classification task for our novel ClusterLoss in the discriminator. The additional adversarial loss, along with our proposed ProjectionLoss, allows our method to generate high quality environment maps for reflection and accurate shading of virtual objects.

3.1. Dataset and processing

In this work we use two datasets: Laval Indoor HDR dataset [10] and PanoContext LDR panoramas [32]. We use the author's test split [10][2] for HDR images, resulting in a total of 2,810 training images. The different sources and scenes have large intensity variations and hence are unsuitable for training as-is. We thus employ a log scale on exposure-normalized HDR linear images for the network. The LDR PanoContext images are only used to train the discriminator. For a given linear HDR ground truth image G_{lin}, we compute the training ground truth G as:

$$G = \min(\max(0, \log_{10}(G_{lin} \cdot \alpha + 1)), 2) - 1, \qquad (1)$$

We empirically choose $\alpha = 0.2 \cdot \overline{G_{lin}}$ to match the middle gray values. The clipping of intensities corresponds to our use of tanh activation for our network, and results in dynamic range of $[0, 100]$ for output linear RGB, given the input LDR images in $[0, 1]$ range.

3.2. Model architecture

Our model, EnvMapNet, consists of an encoder and decoder with skip connections as shown in Fig. 2. Each is composed of building blocks detailed in Appendix 3. The encoder is composed of five sets of EnvMapNet-conv-block and EnvMapNet-downsample-blocks. 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-blocks. The final output is produced by a 3×3 convolution to produce 3 channels for RGB, followed by a tanh activation.

3.3. Image-based losses and ProjectionLoss

We use a weighted combination of image-based and adversarial losses to train our model end-to-end. We want to retain the color for pixels from the known (input) region while hallucinating the rest with a plausible scene completion (typically extrapolation). Guided by the binary mask in the input, we compute an $L1$ loss between the known pixels in the input and corresponding predicted colors. For the completed output, we apply a multi-scale $L2$ loss. This allows the model to coarsely regress the light direction and color, however it does not allow for generation of sharp features for the estimated light sources (as seen in Fig. 1(b) and Fig. 2). To obtain a high contrast HDR result that generates correct shadows $L2$ loss is not sufficient. An ideal solution would be to use a ray tracer to render shadows and penalize the difference between the rendered images using ground truth and predicted maps. This is non-trivial and expensive for end-to-end training.

Considering only shadow casting, the intensity of a pixel in the shadow is dependent on the integral of the environment map except directions blocked by the object. Thus, any pixel on the shadow plane can be approximated by the integral over the masked environment map. We take inspiration from our intuition above, and Wasserstein distance, and introduce our novel ProjectionLoss. We select a set of randomized binary masks P having the same size as our environment map, and create a one-dimensional vector of length $|P|$ (the number of images in the set) by integrating the pixel intensities in corresponding masked images. On a 0-valued background, the masks contain polygons generated with randomization whose height and width range from 10% to 40% of the corresponding image dimensions and filled with value of 1. For further illustration of ProjectionLoss, and examples of the masks used, see Appendix 4 in supplementary materials. The final loss, termed *ProjectionLoss*, is the $L1$ distance between the vectors corresponding to predicted and ground truth environment maps. We show this error using $|P| = 50$ projection masks in Fig. 3(a) for each environment map shown in row (c), with the first

[2] http://vision.gel.ulaval.ca/~jflalonde/projects/deepIndoorLight/test.txt

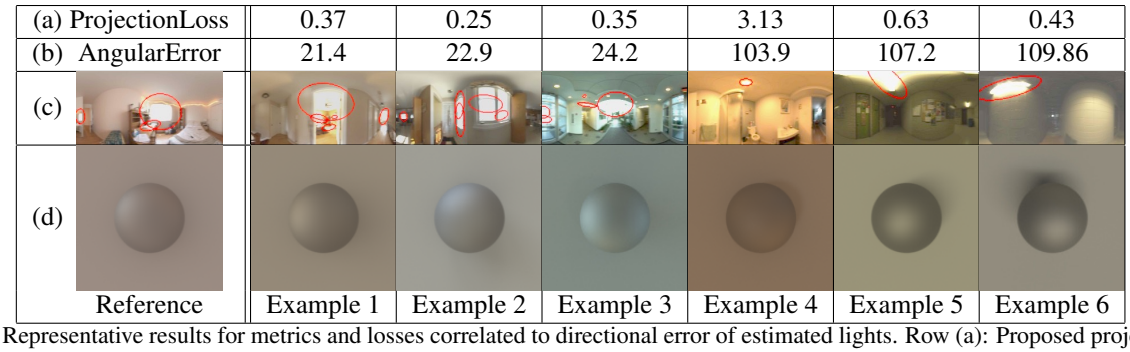

Figure 3. Representative results for metrics and losses correlated to directional error of estimated lights. Row (a): Proposed projection loss on the environment maps w.r.t Reference (see Sec. 3.3). Row (b): Angular errors in degrees, using parametric lights for each environment map w.r.t Reference (see Sec. 4). Row (c): Environment maps with extracted parametric lights shown as red ellipses. Row (d): Rendering of a rough metallic sphere using the corresponding environment map and viewed from top.

image being the reference. In Fig. 3(d) we show rendered spheres using the corresponding environment maps. We can see that lower values of ProjectionLoss correlate with both lower angular error and better visual match of lighting direction to the reference. This is also supported in our ablation study results in Sec. 5.2.

For a ground truth HDR image G, a mask indicating unknown pixels M, a predicted image I, and a set of projection masks P, the final image-based loss is defined as:

$$\begin{aligned} Loss_{projection} &= |\forall_{P_i \in P}(\sum (I * P_i) - \sum (G * P_i))|_1 \\ Loss_{image} &= w_1 |I * M - G * M|_1 + w_2 ||I - G||_2 \\ &\quad + Loss_{projection} \end{aligned} \tag{2}$$

ProjectionLoss is related to diffuse convolution or cosine loss used in previous works [10, 23]. While filtered, or down sampled integral can represent diffuse or low frequency lighting information, we find that our formulation is able to capture high frequency lighting better. To further understand the value of ProjectionLoss for lighting estimation, and to compare with other measures, such as SSIM [33] and Mean Squared Error (MSE), we performed a detailed experiment with user study as described in Appendix 4. We first confirmed that SSIM on rendered images is a good baseline for retrieval of images with similar lighting on objects. We then correlated retrieval of similar environment maps using ProjectionLoss, SSIM and MSE. 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 effectively trains the model for light estimation, such that the end result for rendering is accurate with respect to ground truth. We also show that MSE on the environment map is insufficient for training accurate light estimation. Its use as a metric of comparison, as done in previous works, would not correlate well with the final application.

3.4. Adversarial loss and ClusterLoss

To generate perceptually pleasing completions to the scene, we train the model using a GAN loss. Adversarial training was proposed in [12], where a discriminator was trained to classify samples from the training set as "real" and those produced by a generator as "fake". The generator has a loss component for classification of generated samples as "real". The networks are trained using alternate weight update schedule. Since their advent, there have been many improvements on GANs for training stability and result quality. As discussed previously, many of these have focused on single objects such as faces [16] or increased stability using large datasets or architectures [4]. However, we found that training a mobile-friendly GAN architecture with fewer than $3,000$ images with high appearance variation resulted in mode collapse or non convergence.

It has been shown that providing an additional task to the discriminator using conditional labels or self-supervision can increase stability [21]. Assigning semantic labels (e.g, room types) is non-trivial for panoramic images since the viewpoint could be from between two rooms. Recent work using self-supervision with rotation prediction [5] has been promising. However, for equirectangular images the possible rotations would be flips along the image axis, of which that along the vertical image axis is valid for a scene.

We thus propose a novel scheme combining the learnings from above methods. We begin with the recognition that our application needs to generate "plausibly similar looking images" as the training set, hence the idea to use appearance features to form a secondary task. We use traditional image patch features (mean color and ORB [22]) and assign a K-means-derived cluster ID to each image (we used $K = 5$). Along with the typical real vs. fake classification, the discriminator is supervised to classify the K-means assigned cluster ID for each real image with the proposed ClusterLoss. This allows us to train without additional heuristics or tricks, and avoid mode collapse, as the encoded latents

are forced to spread out over the appearance clusters. Intuitively it also helps the discriminator focus on a low ambiguity appearance-based task that pays attention to spatial locations and details, thus avoiding cases where a single patch artifact can easily clue the discriminator that the image is fake ("easy wins"). The discriminator is composed of residual blocks as detailed in Appendix 3. We use softmax cross entropy loss for the adversarial training of our model and discriminator D. With y, p indicating the ground truth and predicted one-hot encoded vectors for the cluster classification respectively, our adversarial losses are:

$$\begin{aligned} Loss_{cluster} &= \sum_{k=1}^{5} y_{o,k} \log(p_{o,k}) \\ Loss_{fakereal} &= \log(D(G)) + \log(1 - D(I)) \\ Loss_{adversarial} &= -\log(D(I)) \end{aligned} \tag{3}$$

3.5. Implementation and adversarial training

During training we mask the ground truth images using randomly generated polygonal regions, and provide them as corresponding incomplete input environment map. The regions are of irregular shape and can be non-contiguous to generalize our model to complete any partial input. For input, we tonemap the ground truth HDR environment map by dividing the pixels by the average intensity (exposure compensation), applying a gamma with value of 2.2, and final normalization and clipping of the intensity to $[-1.0, 1.0]$. The color channels in the unknown pixels are filled with numbers from uniform random distribution in $[-1.0, 1.0]$ range. We use the logarithmic transform from Sec. 3.1 for the ground truth used in evaluation of losses. The components of $Loss_{image}$ are weighted using $w_1 = 0.5$, $w_2 = 0.01$. For all image-based losses, we also weight the pixels to compensate for the subtended angle on the sphere.

$$\begin{aligned} Loss_{EnvMapNet} &= Loss_{image} + \\ &\quad Loss_{adversarial} + Loss_{cluster} \\ Loss_{discriminator} &= Loss_{fakereal} + Loss_{cluster} \end{aligned} \tag{4}$$

We train the networks by minimizing $Loss_{EnvMapNet}$ and $Loss_{discriminator}$ respectively. We use the Adam optimizer, a batch size of 16, and the learning rate starting at 0.0002, and decayed by half every 50 epochs after the first 100 epochs. We alternate between training the discriminator and EnvMapNet after each mini-batch.

4. Metrics for benchmarking

Visual coherence in AR aims to give a user the illusion that an inserted virtual object belongs in the real scene. However algorithm development, especially in the domain of deep learning, involves many heuristics and parameter tuning. We therefore believe that establishing quantitative metrics, separate from loss functions, that correlate with the target application setting, is important to easily compare different methods. Several previous methods have used MSE on environment maps in RGB, intensity or log formulation as a metric. However, as discussed in Sec. 3.3, it is not well suited to measure accuracy such that the final rendering matches ground truth.

At the core of it we want to solve for two aspects of the environment map: lighting accuracy and perceptual plausibility. Real-time mobile renderers use analytical (point) lights to cast shadows. Hence lighting accuracy is correlated to direction of top-n dominant light sources. Since we cannot (and should not) generate the exact unseen test room for reflection, measures such as MSE or SSIM [33, 29] for one-to-one image comparisons are unsuitable. Instead we need to use plausibility of the hallucinated scene as a metric. We start with a discussion on lighting accuracy metrics followed by quantitative measures for perceptual quality. Our goal is to define metrics that can be used by any future work to objectively compare techniques. Thus we establish a repeatable benchmark that uses the publicly available Laval HDR dataset [10], and reference implementations of the metrics that we will provide.

Methods that can extract light sources from environment maps include median cut [8], variance minimization [28], and parametric fitting from peak finding [9]. In the case of mobile AR, real-time renderers need fast decomposition techniques, hence we use the latter method to find centers and extent of light sources. The angular coordinates of the ellipse center are used to measure the directional accuracy with respect to ground truth. For a given HDR environment map, we iteratively find a seed pixel with maximum intensity, and grow the region connected to the seed till we reach 30% of the peak value. This is repeated until the peak value found is less than 90% of the largest. For each such region, we fit an ellipse covering its convex hull. Example images with parametric light fitting are shown in Fig. 3(c).

To measure the *AngularError* between the ground truth and estimated environment maps, we extract (at most) five parametric lights from each. For each ground truth light we find the minimal angular error to the predicted light set, and vice-versa. We take the mean over the errors from both ground truth lights and predicted lights, as the final AngularError between the two environment maps. Fig. 3 demonstrates the correlation of the AngularError metric, with examples of low and high errors with respect to a selected reference. We can see that environment maps with lower errors produce similar lighting directions on the reference sphere, compared to those with higher errors, thus making it a suitable metric for comparison.

To measure perceptual plausibility, user studies are commonly used for their insight into final experience quality. However, user studies are hard to repeat or compare across publications. We take inspiration from work in GANs, that

(a) (b) (c) (d) (e)

Figure 4. Snapshots from our real-time mobile app. (a) A real and virtual rendering of same object, we can see that the lighting direction and color matches closely. (b) Lighting on the virtual mirror finish sphere is coherent with other real objects in the scene and the reflection formed is also a plausible completion to the scene. (c) A virtual lamp on the table, with its top specular highlight matching the direction on the glass object on the lower right. (d)-(e) Virtual metallic spheres with different roughness. See videos in **supplementary material**.

measure quality of synthesized images using Frechet Inception Distance (FID) [13]. The measure is defined to evaluate similarity of distributions between two image sets, in contrast to perceptual metrics such as SSIM that require a reference image for each given test image. That is, the FID is low between two sets of images that have similar image features, and overall diversity among the different generated samples is the same as in the reference samples. This metric is thus good to demonstrate that not only do the details in any generated environment map match realistic patches from training images, but the set of predicted images also have variety in content without over-fitting.

5. Results

To show the effectiveness of our method for rendering virtual objects in AR, we start with qualitative results from real-world applications of our model, followed by comparisons to previous work and quantitative benchmark results.

5.1. Qualitative results

In Fig. 4 we show results from our prototype iOS app used in real-world scenes. We use device pose and plane geometry provided by ARKit [2] to warp the input camera image into a partial environment map. The completed environment map from running our model on the device is then used with SceneKit[3] to render the virtual object[4]. The inference time is under 9 ms on an iPhone XS, allowing updates at high frame rates depending on desired user experience.

Please see the **supplementary material for videos**[5] from our application, where we use an update rate of 10fps for our results. Fig. 4(a) shows a real cooker in the right, and a rendered version on the left. Note that the material properties of the brushed steel body do not match exactly. We can clearly see that the lighting direction closely matches the reference real-world object, providing the correct specular highlights. The accurate lighting, combined with perceptually plausible reflection clearly makes the rendering an immersive AR experience. In Fig. 4(b) we render a mirror sphere into a scene to show the impact of generating plausible reflections. Though the input observes a very small part of the environment, we are able to create a believable AR experience by completing the environment map with a detailed scene. Even though it cannot match the actual room in every detail, the perceptual impact and coherence of the virtual object with other objects, such as the mirror on the wall, and metallic tower on the right, is evident. In Fig. 4(c)-(e) we provide more examples of inserted virtual objects that have lighting that is coherent with the real objects in the field of view.

From the images in Fig. 1 and 4 we highlight a few aspects of our method. For each of the scenes, that vary in time of day and FoV, not only is the light direction coherent with the scene but so is the type/shape/size of light source generated by our method. Fig. 1(e) shows a virtual teapot on a table illuminated with an unseen artificial light. The roughness of the teapot was 0.2, and metalness was 1.0 in SceneKit. The light direction and highlights from our estimate closely match those on the real teapot (blue) and sphere (pool ball) in the scene. In Fig. 4(b) and (d) the light

[3] https://developer.apple.com/documentation/scenekit/

[4] 3D models from https://developer.apple.com/augmented-reality/quick-look/

[5] https://docs-assets.developer.apple.com/ml-research/papers/hdr-environment-map.mp4

Method	FID	AngularError
EnvMapNet (Ours)	52.7	34.3±18.5
Ours without ProjectionLoss	77.7	39.2±29.9
Ours without ClusterLoss	203	75.1±25
Gardner *et al*. [10]	197.4	65.3±24.5
Artist IBL (Fixed)	-	46.5±15.4

Table 1. Quantitative comparisons and ablation studies (Sec. 5.2).

sources have appearances closer to that of a window/door that are common and plausible in those environments.

We show qualitative comparisons to recent work from Srinivasan *et al*. [25], and their re-implementations of Neural Illumination [23] and Deep Light [17] in **Appendix 5**.

5.2. Quantitative evaluation and benchmarking

For quantitative benchmarking we use the metrics detailed in Sec. 4 on the publicly available Laval dataset [10], and show the results in Table 1. We show some example results in Fig. 5 with rendered objects using results and ground truth (GT) environment maps. All images are best seen on a color monitor with magnification. We highly encourage the reader to see **Appendix 6** for details and more results with renderings.

As discussed in Sec. 2, though several methods have focused on light estimation from single images, only a few generate HDR RGB environment maps. Recent techniques that estimate lighting such as [23] and [17], have not shared code, model weights nor use public datasets. There is also a lack of a standard set of metrics for comparison. The method by Gardner *et al*. [10] is the current state-of-the-art method that is most comparable to our method, uses the same public dataset, and is compared in the above recent works as well. They have provided a web interface to obtain results from their method[6]. Since we use the same splits for train/test, we provide detailed quantitative and qualitative comparisons with their results.

We summarize the quantitative metrics in Table 1, calculated over the 250 test images. The results from using our method are reported as **EnvMapNet**. We introduced two novel loss functions in our method, for which we also conducted ablation studies. For **Ours without ProjectionLoss** we observe higher angular error than our full model EnvMapNet, as expected from our discussions in Sec. 3.3. When we trained **Ours without ClusterLoss**, for the discriminator, the model often suffered from instability as discussed in Sec. 3.4, and could not generate textures with fine details, hence the higher FID and angular error.

The errors for Gardner *et al*. [10] are higher than our complete pipeline. As shown with example images (Fig. 5 and Appendix 6), the environment map generated by their method lacks fine details for the scenes hence the higher

[6] http://rachmaninoff.gel.ulaval.ca:8001/

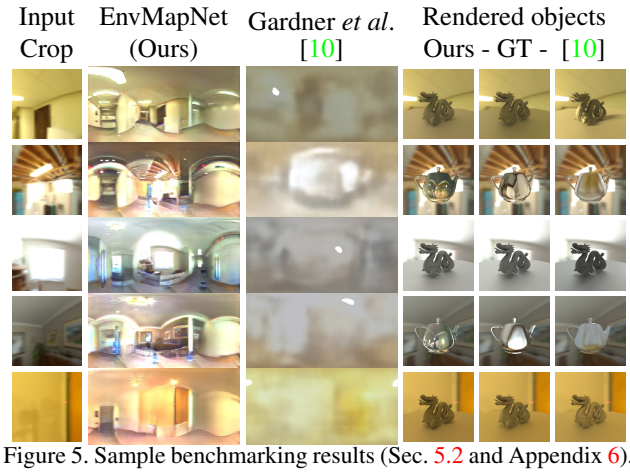

Figure 5. Sample benchmarking results (Sec. 5.2 and Appendix 6).

FID score. The angular error is also higher on average, which could be explained from our experiments in Sec. 3.3 that show MSE loss is insufficient to train for accurate lighting. Both these aspects can be clearly observed to affect the use of the environment maps in rendering objects with both mirror and diffuse finish, along with the cast shadows.

As a baseline measurement, we consider that AR applications, like games, may use a fixed artist-created environment map for lighting. We obtained such an environment map designed for indoor scenes as detailed in Appendix 2. We calculate the angular error metric with respect to this environment map, and report it in Table 1 as **Artist IBL**.

As demonstrated through the various qualitative examples and quantitative benchmarking, our method, with lowest FID and AngularError, can produce high quality environment maps both for visually pleasing reflections, and accurate lighting of the virtual objects.

6. Conclusions

We presented the first method that, given a small field-of-view LDR camera image, can estimate a high resolution HDR environment map in real-time on a mobile device. The result can be used to light objects of any material finish (mirror to diffuse) for augmented reality. We made two novel contributions in the training of our neural network, with ProjectionLoss for the environment map, and ClusterLoss for the adversarial loss. This enabled our method to synthesize environment maps with accurate lighting and perceptually plausible reflections. We proposed two metrics to measure both these aspects of the estimated environment map. Through qualitative and quantitative comparisons we demonstrated that our method reduces the angular error in parametric light direction by more than 50%, along with a 3.7 times reduction in FID. We showcased a real-world mobile application that is able to run our model in real-time (under 9ms) and render visually coherent virtual objects in novel environments.

References

[1] ARCore. *Google LLC*, 2021.

[2] ARKit. *Apple Inc*, 2021.

[3] Dejan Azinović, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *Proc. CVPR*, 2019.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. ICLR*, 2019.

[5] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised GANs via auxiliary rotation loss. In *Proc. CVPR*, 2018.

[6] Dachuan Cheng, Jian Shi, Yanyun Chen, Xiaoming Deng, and Xiaopeng Zhang. Learning scene illumination by pairwise photos from rear and front mobile cameras. *Computer Graphics Forum*, 2018.

[7] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Computer Graphics and Interactive Techniques*, SIGGRAPH, 1998.

[8] Paul Debevec. A median cut algorithm for light probe sampling. In *ACM SIGGRAPH Courses*, 2006.

[9] Marc-Andre Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagne, and Jean-Francois Lalonde. Deep parametric indoor lighting estimation. In *Proc. ICCV*, 2019.

[10] 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.

[11] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-Francois Lalonde. Fast spatially-varying indoor lighting estimation. In *Proc. CVPR*, 2019.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*. 2014.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proc. NIPS*, 2017.

[14] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. 2019.

[15] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *Proc. CVPR*, 2017.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. ICLR*, 2018.

[17] 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.

[18] 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.

[19] Robert Maier, Kihwan Kim, Daniel Cremers, Jan Kautz, and Matthias Nießner. Intrinsic3D: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *Proc. ICCV*, 2017.

[20] Steve Marschner and Donald P. Greenberg. Inverse lighting for photography. In *Color Imaging Conference*, 1997.

[21] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *Proc. ICML*, 2017.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proc. ICCV*, 2011.

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

[24] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proc. CVPR*, 2017.

[25] 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.

[26] Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on Graphics*, 2019.

[27] UnityTechnologies. Reflection probes, 2021.

[28] Kuntee Viriyothai and Paul Debevec. Variance minimization light probe sampling. SIGGRAPH, 2009.

[29] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems Computers*, 2003.

[30] Edward Zhang, Michael F. Cohen, and Brian Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2016.

[31] Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, , Jonathan Eisenmann, and Jean-François Lalonde. All-weather deep outdoor lighting estimation. In *Proc. CVPR*, 2019.

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

[33] 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.