Editable indoor lighting estimation

Henrique Weber[1], Mathieu Garon[2], and Jean-François Lalonde[1]

[1] Université Laval, Québec, Canada
[2] Depix, Montréal, Canada
https://lvsn.github.io/EditableIndoorLight/

Abstract. We present a method for estimating lighting from a single perspective image of an indoor scene. Previous methods for predicting indoor illumination usually focus on either simple, parametric lighting that lack realism, or on richer representations that are difficult or even impossible to understand or modify after prediction. We propose a pipeline that estimates a parametric light that is easy to edit and allows renderings with strong shadows, alongside with a non-parametric texture with high-frequency information necessary for realistic rendering of specular objects. Once estimated, the predictions obtained with our model are interpretable and can easily be modified by an artist/user with a few mouse clicks. Quantitative and qualitative results show that our approach makes indoor lighting estimation easier to handle by a casual user, while still producing competitive results.

Keywords: lighting estimation, virtual object insertion, HDR

1 Introduction

Mixing virtual content realistically with real imagery is required in an increasing range of applications, from special effects to image editing and augmented reality (AR). This has created the need for capturing the lighting conditions of a scene with ever increasing accuracy and flexibility. In his seminal work, Debevec [6] suggested to capture the lighting conditions with a high dynamic range light probe. While it has been improved over the years, this technique, dubbed *image-based lighting*, is still at the heart of lighting capture for high end special effects in movies nowadays[3]. Since the democratization of virtual object insertion for consumer image editing and AR, capturing light conditions with light probes restricts non professional users to have access to the scene and to use specialized equipment. To circumvent those limitations, approaches for automatically estimating the lighting conditions directly from images have been proposed.

In this line of work, the trend has been to estimate more and more *complex* lighting representations. This is exemplified by works such as Lighthouse [25], which propose to learn a multi-scale volumetric representation from an input stereo pair. Similarly, Li et al. [19] learn a dense 2D grid of spherical gaussians over the image plane. Wang et al. [27] propose to learn a 3D volume of similar

[3] See https://www.fxguide.com/fxfeatured/the-definitive-weta-digital-guide-to-ibl/.

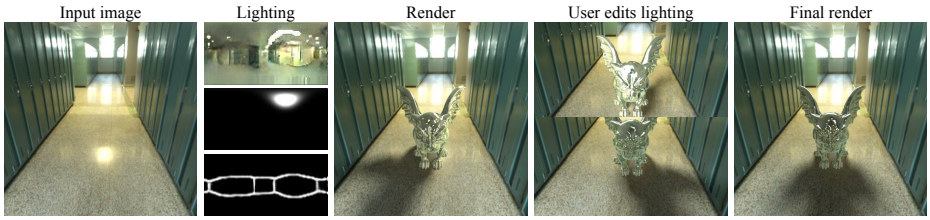


Fig. 1: Our method produces an estimation of the indoor lighting from a single perspective image. Our lighting representation is composed of a 3D parametric light source, a texture map and a coarse 3D layout of the scene. With this information, it is possible to realistically insert 3D objects (like the golden armadillo and sphere) into the scene. Because our lighting representation is interpretable and intuitive, the user can experiment with possibilities by modifying, say, the position of the light source in order to achieve the desired look.

spherical gaussians. While these lighting representations have been shown to yield realistic and spatially-varying relighting results, they have the unfortunate downside of being hard to understand: they do not lend themselves to being easily editable by a user. This quickly becomes a source of limitation when erroneous automatic results need to be corrected for improved accuracy or when creative freedom is required.

In this work, we depart from this trend and propose a *simple, interpretable, and editable* lighting representation (fig. 1). But what does it mean for a lighting representation to be editable? We argue that an editable lighting representation must: 1) *disentangle* various components of illumination; 2) allow an *intuitive control* over those components; and, of course, 3) enable *realistic relighting results*. Existing lighting representations in the literature do not possess all three properties. *Environment maps* [11,24,17] can be rotated but they compound light sources and environment textures together such that one cannot, say, easily increase the intensity of the light source without affecting everything else. Rotating the environment map inevitably rotates the entire scene, turning walls into ceilings, etc., when changing the elevation. *Dense and/or volumetric* representations [12,19,25,27] are composed of 2D (or 3D) grids containing hundreds of parameters, which would have to be modified in a consistent way to achieve the desired result, an unachievable task for most. *Parametric* representations [10] model individual light sources with a few intuitive parameters, which can be modified independently of the others, but cannot generate realistic reflections.

Our proposed representation is the first to offer all three desired properties and is composed of two parts: 1) a parametric light source for modeling shading in high dynamic range; and 2) a non-parametric texture to generate realistic reflections off of shiny objects. Our representation builds on the hypothesis (which we validate) that most indoor scenes can accurately be modeled by a *single*, dominant directional light source. We model this in high dynamic range with a parametric representation [10] that explicitly models the light source intensity,

size, and 3D position. This representation is intuitive and can easily be edited by a user simply by moving the light source around in 3D.

This light source is complemented with a spatially-varying environment map texture, mapped onto a coarse 3D representation of the indoor scene. For this, we rely on a layout estimation network, which estimates a cuboid-like model of the scene from the input image. In addition, we also use a texture estimation network, whose output is conditioned on a combination of the input image, the scene layout and the parametric lighting representation. By explicitly tying the appearance of the environment texture with the position of the parametric light source, modifying the light source parameters (e.g. moving around the light) will automatically adjust the environment in a realistic fashion.

While our representation is significantly simplified, we find that it offers several advantages over the previous approaches. First, it renders both realistic shading (due to the high dynamic range of the estimated parametric light) and reflections (due to the estimated environment map texture). Second, it can efficiently be trained on real images, thereby alleviating any domain gap that typically arise when approaches need synthetic imagery for training [25,19,27]. Third—and perhaps most importantly—it is *interpretable and editable*. Since all automatic approaches are bound to make mistakes, it is of paramount importance in many scenarios that their output be adjustable by a user. By modifying the light parameters and/or the scene layout using simple user interfaces, our approach bridges the gap between realism and editability for lighting estimation.

2 Related work

For succinctness, we focus on single-image indoor lighting estimation methods in the section below, and refer the reader to the recent survey on deep models for lighting estimation for a broader overview [8].

Lighting estimation Gardner et al. [11] proposed the first deep learning-based lighting estimation method for indoor scenes, and predicted an HDR environment map (equirectangular image) from a single image. This representation was also used in [17] for both indoors and outdoors, in [24] to take into account the object insertion position, in [23] which presented a real-time on-device approach, in [22] for scene decomposition, and in [3] which exploited the front and back cameras in current mobile devices. Finally, [28] propose to learn the space of indoor lighting using environment maps on single objects.

Other works explored alternative representations, such as spherical harmonics [12,20,34] that are useful for real-time rendering but are typically unsuitable for modeling high-frequency lighting (such as bright light sources) and are not ideal for non diffuse object rendering. [10] proposed to estimate a set of 3 parametric lights, which can easily be edited. However, that representation cannot generate realistic reflections. EMlight [33] propose a more expressive model by predicting gaussians on a spherical model. Similar to us, GMlight [31] back-projects the spherical gaussians to an estimated 3D model of the scene. This is

further extended in [1] by the use of graph neural networks, and in [32] through the use of spherical wavelets dubbed "needlets".

Recently, methods have attempted to learn volumetric lighting representations from images. Of note, Lighthouse [25] learns multi-scale volumetric lighting from a stereo pair, [19] predicts a dense 2D grid of spherical gaussians which is further extended into a 3D volumetric representation by Wang et al. [27]. While these yield convincing spatially-varying results, these representations cannot easily be interacted by a user.

Scene decomposition Holistic scene decomposition [2] is deeply tied to lighting estimation as both are required to invert the image formation process. Li et al. [19] proposes to extract the scene geometry and the lighting simultaneously. Similarly, [7] extract only the geometry of the scene by estimating the normal and depth of the scene. These geometric representations are however non-parametric and thus difficult to edit or comprehend. [16] proposes a simplified parametric model where a room layout is recovered in the camera field of view. Similarly, [35] presents a method to estimate the layout given a panoramic image of an indoor scene. We use the method of [16] to estimate a panoramic layout given a perspective image, thus providing a simple cuboid representation that allows for spatially varying textured lighting representation.

3 Editable indoor lighting representation

We begin by presenting our hybrid parametric/non-parametric lighting representation which aims at bridging the gap between realism and editability. We also show how that representation can be fitted to high dynamic range panoramas to obtain a training dataset, and conclude by presenting how it can be used for virtual object relighting.

3.1 Lighting representation

Our proposed light representation, shown in fig. 2, is composed of two main components: an HDR parametric light source $\mathbf{p}$; and an LDR textured cuboid $\mathcal{C}$.

Light source As in [10], the light source parameters $\mathbf{p}$ are defined as

$$\mathbf{p} = \{\mathbf{l}, d, s, \mathbf{c}, \mathbf{a}\}\,, \tag{1}$$

where $\mathbf{l} \in \mathbb{R}^3$ is a unit vector specifying the light direction in XYZ coordinates, d is the distance in meters, s the radius (in meters), $\mathbf{c}, \mathbf{a} \in \mathbb{R}^3$ are the light source and ambient colors in RGB, respectively. Here, $\mathbf{l}$, d and s are defined with respect to the camera. In contrast with [10], we use a single light source.

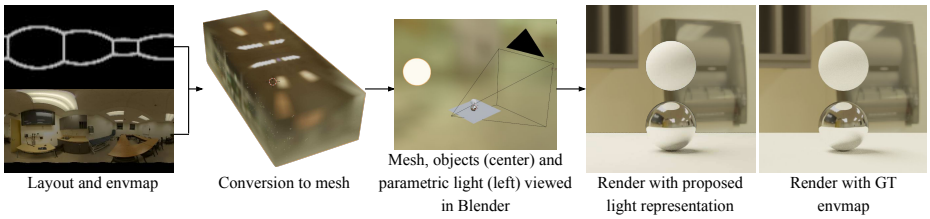


Fig. 2: To render a virtual object with our proposed lighting representation, the texture is first warped according to the layout (1st column), producing a textured mesh (2nd). This mesh is combined with an emitting sphere representing the parametric light (3rd) for rendering. The resulting rendering (4th) closely matches the ground truth rendering obtained with the HDR environment map (last).

Textured cuboid The cuboid $\mathcal{C} = \{\mathbf{T}, \mathbf{L}\}$ is represented by a texture $\mathbf{T} \in \mathbb{R}^{2H \times H \times 3}$, which is an RGB spherical image of resolution $2H \times H$ stored in equirectangular (latitude-longitude) format, and a scene layout $\mathbf{L} \in \mathbb{R}^{2H \times H}$. The layout is a binary image of the same resolution, also in equirectangular format, indicating the intersections of the main planar surfaces in the room (walls, floor, ceiling) as an edge map [9].

3.2 Ground truth dataset

The ground truth is derived from the Laval Indoor HDR Dataset [11], which contains 2,100 HDR panoramas (with approximate depth labels from [10]). We extract $\mathbf{p}$ and $\mathcal{C}$ from each panorama using the following procedure. First, the HDR panorama is clipped to LDR (we re-expose such that the 90th-percentile is 0.8 then clip to [0, 1]) and directly used as the texture $\mathbf{T}$. Then the intersection between the main surfaces are manually labelled to define the layout $\mathbf{L}$. Lastly, we extract a dominant parametric light source from the HDR panorama. In order to determine the main light source, the $N = 5$ brightest individual light sources are first detected using the region-growing procedure in [10]. A test scene (9 diffuse spheres arranged in a 3×3 grid on a diffuse ground plane, seen from top as in fig. 4b) is rendered with each light source independently by masking out all other pixels—the brightest render determines the strongest light source.

An initial estimate of the light parameters $\mathbf{p}$ are obtained by the following. The distance d is approximated by using the average depth of the region, direction $\mathbf{l}$ as the region centroid, the angular size from the major and minor axes of an ellipse fitted to the same region. Finally, the light color $\mathbf{c}$ and ambient term $\mathbf{a}$ are initialized with a least-squares fit to a rendering of the test scene using the HDR panorama. From the initial parameters, $\mathbf{p}$ is further refined:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} ||\mathcal{R}(\mathbf{p}) - \mathcal{R}(\tilde{\mathbf{P}})||_2 \,. \tag{2}$$

$\mathcal{R}(x)$ is a differentiable rendering operator (implemented with Redner [18]) that renders a test scene using $\mathbf{p}$. The optimization is performed using gradient descent

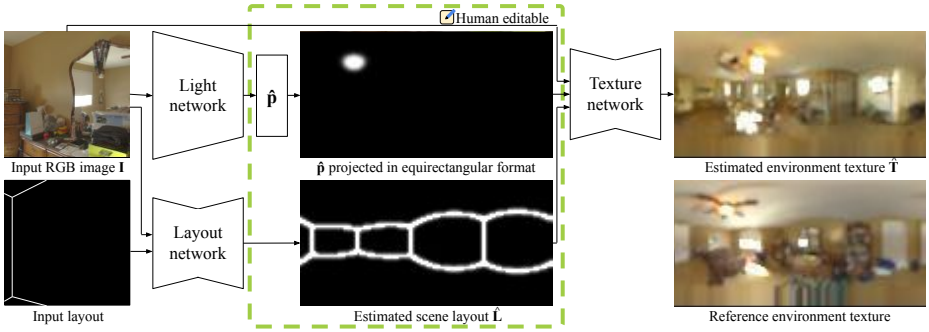


Fig. 3: Our method takes as input a perspective, RGB image and its scene layout representation, passes the RGB to a CNN to predict a parametric light, and passes the partial layout to another CNN to predict the full panorama layout. The parametric light is converted to a binary mask panorama, which is then sent together with the full layout prediction and the input RGB image to a third network which outputs an LDR texture with the light at the desired location.

with Adam [15]. Finally, the texture map $\mathbf{T}$ is rescaled with the estimated ambient term $\mathbf{a}^*$ to ensure that the texture yields the same average RGB color.

3.3 Virtual object rendering

To render a virtual object using our lighting representation, we employ the Cycles rendering engine[4]. A scene, as shown in fig. 2, is composed of a 3D emissive sphere for the parametric light $\mathbf{p}$ and the textured cuboid mesh $\mathcal{C}$. The cuboid mesh is derived by detecting the cuboid corners from the layout using high pass filters. We use the following geometric constraints to simplify the back-projection of the scene corners to 3D. First, the shape is limited to a cuboid, meaning that opposing faces are parallel. Second, the panorama layouts were trained using a camera elevation of 0° (pointing at the horizon) and height of 1.6 meter above the ground. Using these constraints, the bottom corners can easily be projected on the ground plane, and the top corners can be used to compute the ceiling height (averaged from the 4 corners). A texture map can then be computed using every planar surfaces of the cuboid. Finally, the parametric light and the texture are rendered in two rendering passes. After rendering, the relit virtual object can be composited into the image using differential rendering [6].

4 Approach

Our approach, illustrated in fig. 3, is composed of three main networks: light, layout, and texture which are combined together to estimate our light represen-

[4] Available within Blender at `https://www.blender.org`.

tation (c.f., sec. 3) from an image. We assume that the layout of the input image is available, in practice this is obtained with an off-the-shelf solution [30].

Light network A "light" network is trained to learn the mapping from input image $\mathbf{I} \in \mathbb{R}^{128\times128\times3}$ to estimated lighting parameters $\mathbf{p}$ (sec. 3) using a similar approach to [10]. Specifically, the light network is composed of a headless DenseNet-121 encoder [14] to produce a 2048-dimensional latent vector, followed by a fully-connected layer (512 units), and ultimately with an output layer producing the light source parameters $\mathbf{p}$.

The light network is trained on light parameters fitted on panoramas from the Laval Indoor HDR Dataset [11] using the procedure described in sec. 3.2. To generate the input image from the panorama, we follow [11] and extract rectified crops from the HDR panoramas. The resulting images are converted to LDR by re-exposing to make the median intensity equal to 0.45, clipping to 1, and applying a $\gamma = 1/2.4$ tonemapping. The same exposure factor is subsequently applied to the color $\mathbf{c}$ and ambient a light parameters to ensure consistency. Note that the training process is significantly simplified compared to [11] as the network predicts only a single set of parameters.

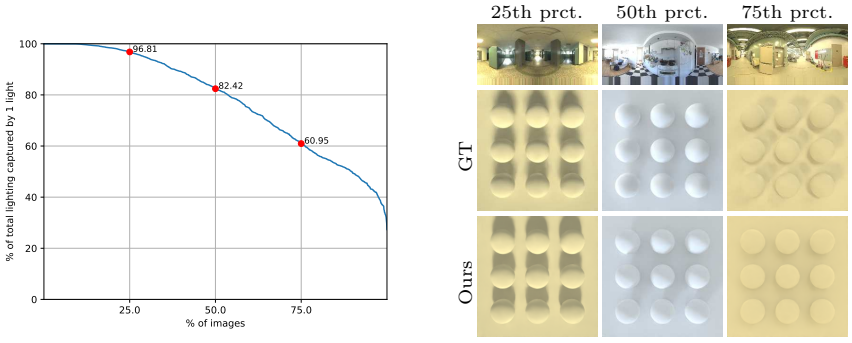
We employ individual loss functions on each of the parameters independently: L2 for direction $\mathbf{l}$, depth d, size s, and ambient color a, and L1 for light color $\mathbf{c}$. In addition, we also employ an angular loss for both the ambient and light colors a and $\mathbf{c}$ to enforce color consistency. The weights for each term were obtained through a Bayesian optimization on the validation set (see supp. mat.).

Layout network The mapping from the input RGB image $\mathbf{I}$ and its layout (obtained with [30]) to the estimated scene layout $\hat{\mathbf{L}}$ (sec. 3) is learned by the "layout" network whose architecture is that of pix2pixHD [26]. Both inputs are concatenated channel-wise. The layout network is trained on both the Laval and the Zillow Indoor Dataset [5], which contains 67,448 LDR indoor panoramas of 1575 unfurnished residences along with their scene layouts. To train the network, a combination of GAN, feature matching and perceptual losses are employed [26]. The same default weights as in [26] are used in training.

Texture network Finally, the estimated environment texture $\hat{\mathbf{T}}$ is predicted by a "texture" network whose architecture is also that of pix2pixHD [26]. It accepts as input a channel-wise concatenation of three images: the input RGB image $\mathbf{I}$, the estimated light parameters $\hat{\mathbf{p}}$ projected in an equirectangular format, and the estimated scene layout $\hat{\mathbf{L}}$. The equirectangular images are vertically concatenated to the input image. Note that the $\hat{\mathbf{p}}$ projection is performed using a subset of all parameters (direction $\mathbf{l}$ and size s only).

The texture network is also trained on both Laval and Zillow datasets. To obtain the required light source position from the Zillow dataset, we detect the largest connected component whose intensity is above the 98th percentile over the upper half of the panorama. To convert the Laval HDR panoramas to LDR,

first a scale factor is found such as the crop taken from that panorama has its 90th percentile mapped to 0.8. This scale factor is then applied to the panorama such as its scale matches the one of the crop. The texture network is trained with the same combination of losses as the layout network.



(a) (b)

Fig. 4: Validation of our 1-light approximation. (a) Cumulative distribution of the contribution of the single strongest light with respect to the entire lighting environment of the scene. (b) Example images for different percentiles, where the rows correspond to the environment map (top), a synthetic scene (seen from the top) rendered with (middle) the ground truth environment map and (bottom) our 1-light representation. As expected, scenes where the strongest light does not contribute significantly have shadows that are less pronounced which may point to several light sources equally contributing to the overall energy (25th prct.). The strongest light source contributes to more than 80% of the total energy in at least 50% of the images in our test set, which confirms our assumption that most scenes can accurately be modeled with a single light source.

5 Experiments

5.1 Validation of our 1-light approximation

We test our hypothesis that most indoor scenes are well-approximated by a single dominant light source with an ambient term. We render a scene with the ground truth environment map, and compare it with the renders obtained from the parametric lighting optimization procedure described in sec. 3. Fig. 4a shows the cumulative distribution of the contribution of the strongest light with respect to the entire lighting of the scene. Note that the strongest light source contributes to more than 95%/80%/60% of the total lighting for 25%/50%/75% of the images in our test set. Fig. 4b shows example images for each of these scenarios. We

Table 1: Quantitative comparative metrics on (left) renderings of a diffuse scene, and (right) on the estimated environment maps directly. Each row is color-coded as best and second best. We also highlight the methods which produce lighting representations that can be interpreted and edited by a user ("Edit.").

	si-RMSE$_\downarrow$	RMSE$_\downarrow$	RGB ang.$_\downarrow$	PSNR$_\uparrow$	FID$_\downarrow$	Edit.
Ours	0.081	0.209	4.13°	12.79	89.58	yes
Gardner'19 (1) [10]	0.099	0.229	4.43°	12.25	356.8	yes
Gardner'19 (3) [10]	0.105	0.508	4.58°	10.87	335.6	yes
Gardner'17 [11]	0.123	0.628	8.29°	10.24	254.8	no
Garon'19 [12]	0.096	0.254	8.04°	9.70	314.9	no
Lighthouse [25]	0.120	0.253	14.53°	9.88	195.5	no
EMLight [33]	0.099	0.232	3.99°	10.38	121.09	no
EnvmapNet[5] [23]	0.097	0.286	7.67°	11.74	201.20	no

find that even if we expect indoor scenes to have multiple light sources, the vast majority can accurately be represented by a single *dominant* light.

5.2 Light estimation comparison

We now evaluate our method and compare it with recent state-of-the-art light estimation approaches. We first validate that our model performs better on quantitative metrics evaluated on physic-based renders of a scene using a test set provided by [11]. For each of the 224 panoramas in the test split, we extract 10 images using the same sampling distribution as in [11], for a total of 2,240 images for evaluation. We also show renders in various scenes to demonstrate how our solution is visually more appealing.

Quantitative comparison To evaluate the lighting estimates, we render a test scene composed of an array of spheres viewed from above (sec. 3) and compute error metrics on the resulting rendering when compared to the ground truth obtained with the original HDR panorama. We report RMSE, si-RMSE [13], PSNR, and RGB angular error [17]. We also compute the FID[6] on the resulting environment maps to evaluate the realism of reflections (similar to [23]).

We evaluate against the following works. First, two versions of [10] are compared: the original (3) where 3 light sources are estimated, and a version (1) trained to predict a single parametric light. Second, we also compare to Lighthouse [25], which expects a stereo pair as input. As a substitute, we generate a second image with a small baseline using Synsin [29] (visual inspection confirmed this yields reasonable results). For [12], we select the coordinates of the image center for the object position. For [23], we implemented their proposed "Cluster

[5] Only their proposed ClusterID loss and tonemapping.
[6] Implementation taken from `https://pypi.org/project/pytorch-fid/`.

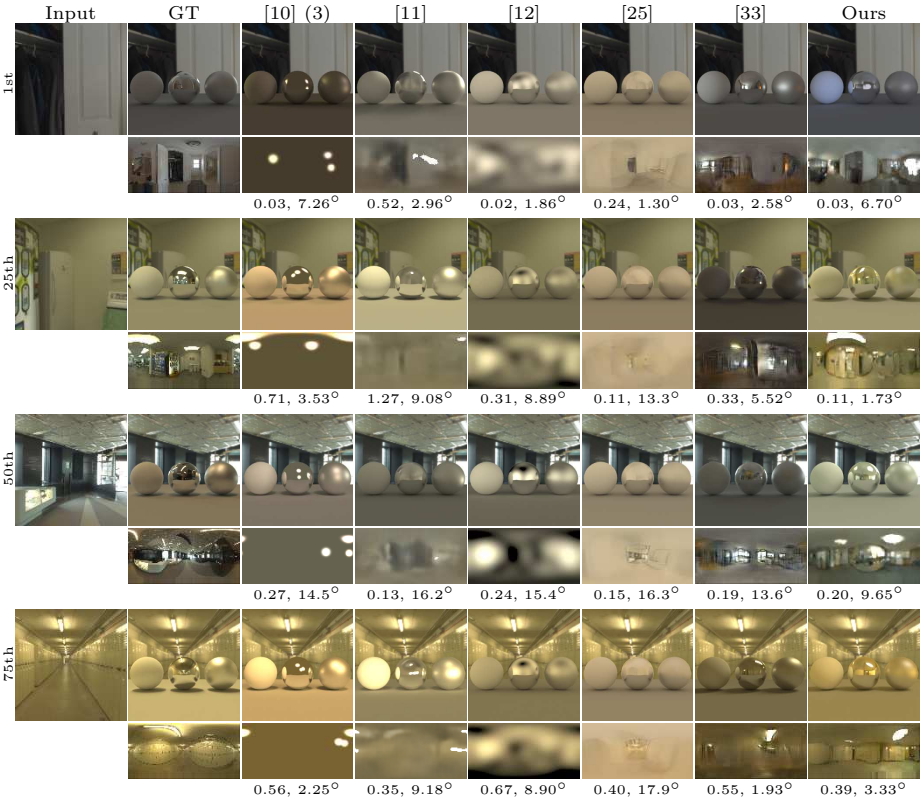


Fig. 5: Qualitative lighting estimation examples from our test set. To compare the estimated lighting, we render a simple scene composed of three spheres (diffuse, mirror, glossy) on a diffuse ground plane with different methods. From left to right: input image, ground truth lighting, Gardner'19 [10] (3 lights), Gardner'17 [11], Garon'19 [12], Lighthouse [25], EMLight [33], and ours. The second row shows the corresponding estimated lighting in equirectangular format (reprojected in the center of the image for the spatially-varying techniques such as [10,12,25] and ours). Finally, error metrics (RMSE and RGB angular) are also shown below each example for reference. Each group shows examples from different error percentiles for our method according to the RMSE metric. More examples can be found in the supplementary materials.

ID loss" and tonemapping (eq. 1 in [23]) but used pix2pixHD as backbone. Finally, we also compare against [33]. Results for each metrics are reported in tab. 1, which shows that despite our model being simple, it achieves the best score in every metric. We argue that it is *because* of its simplicity that we can achieve competitive results. Our approach can be trained on real data (as opposed to [12,25] and does not require an elaborate 2-stage training process (compared to

Fig. 6: Virtual object insertion in scenes with our estimated lighting. For simplicity, we assume the scene surrounding the objects is made of a flat ground plane, which catches shadows and is placed manually by an artist (the focus of our work being lighting estimation). For example, the figure shows a golden armadillo and sphere inserted into three different scenes. Note how the reflections on the objects and the shadows cast on the ground plans appear realistic.

[10]). We also demonstrate a significantly lower FID score than other methods thus bridging the gap between representation realism and HDR accuracy.

Qualitative comparison We also present qualitative results in fig. 5 where predictions are rendered on 3 spheres with varying reflectance properties (diffuse, mirror, glossy). In addition, a tonemapped equirectangular view of the estimated light representation is provided under each render. We show an example from each error percentiles according to the RMSE metric. Our proposed method is perceptually better on the mirror spheres as other methods do not model high frequency details from the scene. We also notice accurate shadow and shading from all the spheres. We show objects realistically composited into photographs in fig. 6. Note how the reflections of the virtual objects and their cast shadows on the ground plane perceptually match the input photograph. Finally, we also compare against [27] in fig. 7.

5.3 Ablation study on input layout

One may consider that requiring the image layout as input may make our method sensitive to its estimation. To show this is not the case, we perform an experiment where we provide a black layout as input to the layout network (equivalent to

(a) Input image (b) Wang et al. (c) Ours

Fig. 7: Qualitative comparison against Wang et al. [27]

no layout estimate). As can be seen in fig. 8, providing a black layout as input simply results in a different layout prediction where the texture still remains coherent with the RGB input and estimated light direction. The FID of the generated panoramas with no input layout is 88.68 (compared to 89.58 from tab. 1), showing that this essentially has no impact.

5.4 Ablation study on the texture network

We also tested different configurations for the texture network in order to validate our design choices. More specifically, we trained the texture network providing as input: (1) only the RGB crop (FID of 167.39), (2) RGB crop and parametric light (FID of 97.13), and (3) RGB crop and layout (FID of 151.04). In contrast, our full approach obtained an FID of 89.57 (see tab. 1).

6 Editing the estimated lighting

Because of its intuitive nature, it is simple and natural for a user to edit our estimated lighting representation, should the estimate not perfectly match the background image or simply for artistic purposes. Fig. 9 shows that our approach simultaneously *disentangles* various components of illumination, allows an *intuitive control* over those components, and enables *realistic relighting results*. First, fig. 9a shows that a user can rotate the light source about its azimuth angle. Note how the estimated texture (second row) is consistent with the desired light position (third row), while preserving the same overall structure. The renders (first row) exhibit realistic reflections and shadows that correspond to the desired lighting directions. A similar behaviour can be observed in figs 9b and 9c when the elevation angle and size are modified, respectively. In fig. 9d, we show that it is also possible to edit the scene layout and obtain an estimated texture map $\hat{\mathbf{T}}$ that is consistent with the users request. We also show results of compositing virtual objects directly into a scene in fig. 6. As shown in fig. 1, realistic rendering

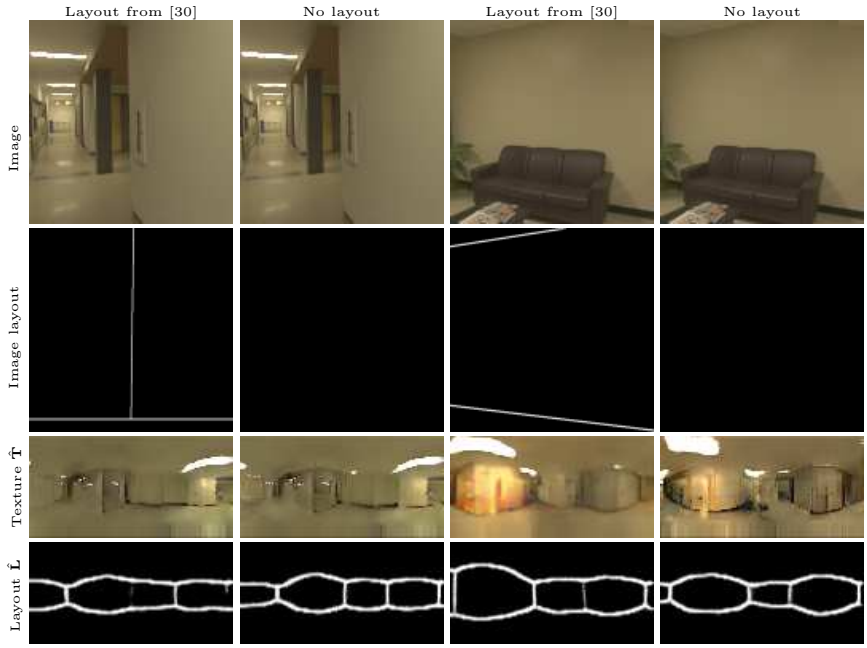


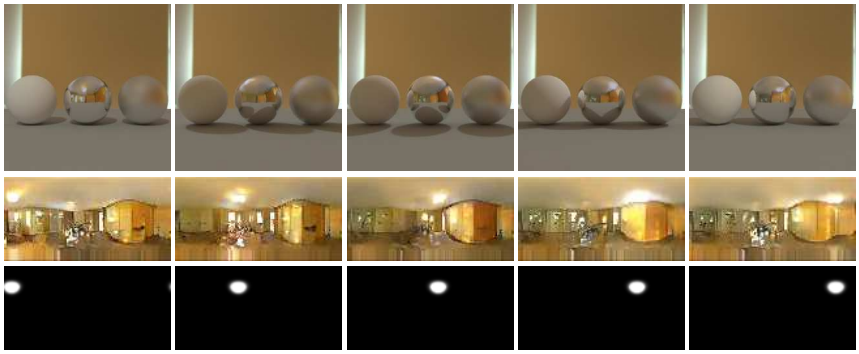
Fig. 8: Ablation on input image layout. We compare the output of our method (last two rows) as a function of whether or not it is given the estimated layout of the input image (with [30]). Our approach produces similar results in both cases.

results can intuitively be edited to achieve the desired look. To the best of our knowledge, the only other method which allows intuitive editing of indoor lighting estimate is that of Gardner et al. [10]. Unfortunately, realistic renders are limited to diffuse objects and cannot be extended to reflective objects as shown in fig. 5.

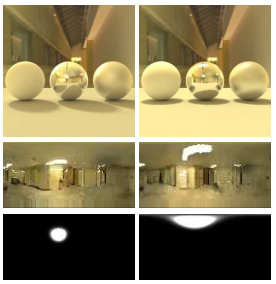
7 Discussion

This paper proposes a lighting estimation approach which produces an intuitive, user-editable lighting representation given a single indoor input image. By explicitly representing the dominant light source using a parametric model, and the ambient environment map using a textured cuboid, our approach bridges the gap between generating realistic shading (produced by HDR light sources) and reflections (produced by textured environment maps) on rendered virtual objects. We demonstrate, through extensive experiments, that our approach provides competitive quantitative performance when compared to recent lighting estimation techniques. In particular, when compared to the only other approach which can be user-edited [10], our approach yields significant improved results.

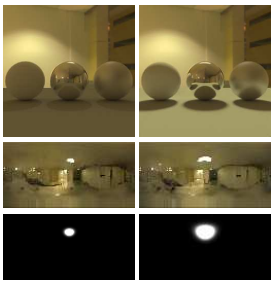
Limitations and future work While our proposed approach estimates a 3D representation of the surrounding lighting environment, it does not reason about



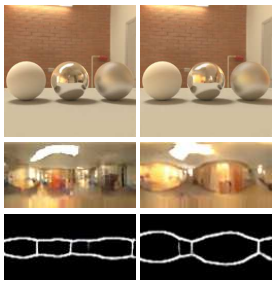
(a) Light azimuth



(b) Light elevation



(c) Light size



(d) Scene layout

Fig. 9: Using our representation, a user can easily edit the estimated light parameters and obtain relighting results consistent with their edits. For example, the user can change the (a) azimuth and (b) elevation angles of the light source; (c) the size of the light source; or (d) the layout of the scene. For all scenarios, we show rendered virtual objects in the first row, the estimated texture $\hat{\mathbf{T}}$ in the second, and the representation being edited in the last (light parameters $\hat{\mathbf{p}}$ for (a)–(c) and layout $\hat{\mathbf{L}}$ for (d)).

light occlusions in the scene as opposed to other techniques such as [12,19,27]. Incorporating these higher-order interactions while maintaining interpretability and editability of the output representation is an interesting direction for future research. In addition, the estimated environment textures were shown to produce realistic reflections on shiny objects, but a close inspection reveals that they are low resolution and contain some visual artifacts. It is likely that more recent image-to-image translation architectures [4,21] could be used to improve realism.

Acknowledgements This research was supported by MITACS and the NSERC grant RGPIN-2020-04799. The authors thank Pascal Audet for his help.

References

1. Bai, J., Guo, J., Wan, C., Chen, Z., He, Z., Yang, S., Yu, P., Zhang, Y., Guo, Y.: Deep graph learning for spatially-varying indoor lighting prediction. arXiv preprint arXiv:2202.06300 (2022)
2. Barron, J.T., Malik, J.: Shape, illumination, and reflectance from shading. IEEE TPAMI **37**(8), 1670–1687 (2014)
3. Cheng, D., Shi, J., Chen, Y., Deng, X., Zhang, X.: Learning scene illumination by pairwise photos from rear and front mobile cameras. Computer Graphics Forum **37**(7), 213–221 (2018)
4. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: CVPR (2020)
5. Cruz, S., Hutchcroft, W., Li, Y., Khosravan, N., Boyadzhiev, I., Kang, S.B.: Zillow indoor dataset: Annotated floor plans with 360º panoramas and 3d room layouts. In: CVPR (2021)
6. Debevec, P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. pp. 189–198. SIGGRAPH (1998)
7. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision. pp. 2650–2658 (2015)
8. Einabadi, F., Guillemaut, J.Y., Hilton, A.: Deep neural models for illumination estimation and relighting: A survey. Computer Graphics Forum **40**(6), 315–331 (2021)
9. Fernandez-Labrador, C., Facil, J.M., Perez-Yus, A., Demonceaux, C., Civera, J., Guerrero, J.J.: Corners for layout: End-to-end layout recovery from 360 images. IEEE Robotics and Automation Letters **5**(2), 1255–1262 (2020)
10. Gardner, M.A., Hold-Geoffroy, Y., Sunkavalli, K., Gagne, C., Lalonde, J.F.: Deep parametric indoor lighting estimation. In: ICCV (2019)
11. Gardner, M.A., Sunkavalli, K., Yumer, E., Shen, X., Gambaretto, E., Gagné, C., Lalonde, J.F.: Learning to predict indoor illumination from a single image. ACM TOG **36**(6) (2017)
12. Garon, M., Sunkavalli, K., Hadap, S., Carr, N., Lalonde, J.F.: Fast spatially-varying indoor lighting estimation. In: CVPR (2019)
13. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 2335–2342. IEEE (2009)
14. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Lee, C.Y., Badrinarayanan, V., Malisiewicz, T., Rabinovich, A.: Roomnet: End-to-end room layout estimation. In: ICCV (2017)
17. LeGendre, C., Ma, W.C., Fyffe, G., Flynn, J., Charbonnel, L., Busch, J., Debevec, P.: Deeplight: Learning illumination for unconstrained mobile mixed reality. In: CVPR (2019)
18. Li, T.M., Aittala, M., Durand, F., Lehtinen, J.: Differentiable monte carlo ray tracing through edge sampling. ACM TOG **37**(6), 1–11 (2018)

19. Li, Z., Shafiei, M., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In: CVPR (2020)
20. Mandl, D., Yi, K.M., Mohr, P., Roth, P., Fua, P., Lepetit, V., Schmalstieg, D., Kalkofen, D.: Learning lightprobes for mixed reality illumination. In: ISMAR (2017)
21. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: ECCV (2020)
22. Sengupta, S., Gu, J., Kim, K., Liu, G., Jacobs, D.W., Kautz, J.: Neural inverse rendering of an indoor scene from a single image. In: ICCV (2019)
23. Somanath, G., Kurz, D.: Hdr environment map estimation for real-time augmented reality. In: CVPR (2021)
24. Song, S., Funkhouser, T.: Neural illumination: Lighting prediction for indoor environments. In: CVPR (2019)
25. Srinivasan, P.P., Mildenhall, B., Tancik, M., Barron, J.T., Tucker, R., Snavely, N.: Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In: CVPR (2020)
26. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: CVPR (2018)
27. Wang, Z., Philion, J., Fidler, S., Kautz, J.: Learning indoor inverse rendering with 3d spatially-varying lighting. In: ICCV (2021)
28. Weber, H., Prévost, D., Lalonde, J.F.: Learning to estimate indoor lighting from 3d objects. In: 3DV (2018)
29. Wiles, O., Gkioxari, G., Szeliski, R., Johnson, J.: Synsin: End-to-end view synthesis from a single image. In: CVPR (2020)
30. Yang, C., Zheng, J., Dai, X., Tang, R., Ma, Y., Yuan, X.: Learning to reconstruct 3d non-cuboid room layout from a single rgb image. In: Wint. Conf. App. Comp. Vis. (2022)
31. Zhan, F., Yu, Y., Wu, R., Zhang, C., Lu, S., Shao, L., Ma, F., Xie, X.: Gmlight: Lighting estimation via geometric distribution approximation. IEEE TIP (2022)
32. Zhan, F., Zhang, C., Hu, W., Lu, S., Ma, F., Xie, X., Shao, L.: Sparse needlets for lighting estimation with spherical transport loss. In: ICCV (2021)
33. Zhan, F., Zhang, C., Yu, Y., Chang, Y., Lu, S., Ma, F., Xie, X.: Emlight: Lighting estimation via spherical distribution approximation. In: AAAI (2021)
34. Zhao, Y., Guo, T.: Pointar: Efficient lighting estimation for mobile augmented reality. In: ECCV (2020)
35. Zou, C., Colburn, A., Shan, Q., Hoiem, D.: Layoutnet: Reconstructing the 3d room layout from a single rgb image. In: CVPR (2018)