Physically-Based Editing of Indoor Scene Lighting from a Single Image

Zhengqin Li[1(✉)], Jia Shi[1,3], Sai Bi[1,2], Rui Zhu[1], Kalyan Sunkavalli[2], Miloš Hašan[2], Zexiang Xu[2], Ravi Ramamoorthi[1], and Manmohan Chandraker[1]

[1] UC San Diego, San Diego, USA
lizhengqin2012@gmail.com
[2] Adobe Research, San Jose, USA
[3] Carnegie Mellon University, Pittsburgh, USA

Abstract. We present a method to edit complex indoor lighting from a single image with its predicted depth and light source segmentation masks. This is an extremely challenging problem that requires modeling complex light transport, and disentangling HDR lighting from material and geometry with only a partial LDR observation of the scene. We tackle this problem using two novel components: 1) a holistic scene reconstruction method that estimates reflectance and parametric 3D lighting, and 2) a neural rendering framework that re-renders the scene from our predictions. We use physically-based light representations that allow for intuitive editing, and infer both visible and invisible light sources. Our neural rendering framework combines physically-based direct illumination and shadow rendering with deep networks to approximate global illumination. It can capture challenging lighting effects, such as soft shadows, directional lighting, specular materials, and interreflections. Previous single image inverse rendering methods usually entangle lighting and geometry and only support applications like object insertion. Instead, by combining parametric 3D lighting estimation with neural scene rendering, we demonstrate the first automatic method for full scene relighting from a single image, including light source insertion, removal, and replacement.

1 Introduction

Light sources of various shapes, colors and types, such as lamps and windows, play an important role in determining indoor scene appearances. Their influence leads to several interesting phenomena such as light shafts through an open window on a sunlit day, highlights on specular surfaces due to incandescent lamps, interreflections from colored walls, or shadows cast by furniture in the room. Correctly attributing those effects to individual visible or invisible light sources in a single image enables abilities for photorealistic augmented reality that have previously been intractable—virtual furniture insertion under varying illuminations with consistent highlights and shadows, virtual try-on of wall paints

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20068-7_32.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13666, pp. 555–572, 2022.
https://doi.org/10.1007/978-3-031-20068-7_32

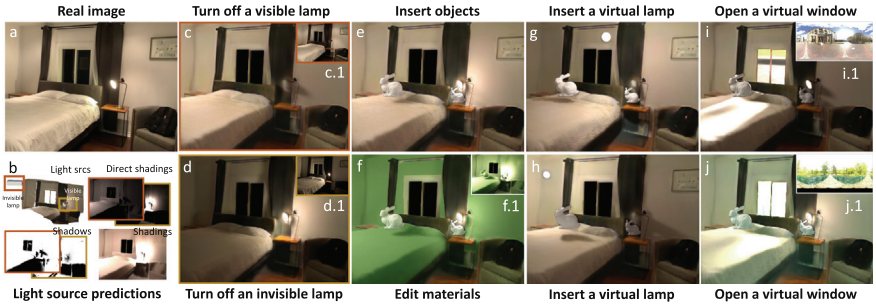

Fig. 1. We present the first method for globally consistent editing of indoor lighting from a single LDR image. Given the input (a), our framework first estimates physically-based light source parameters, for both visible and invisible lights, and then renders their direct contributions and interreflections through a neural rendering framework (b). Our framework can turn off visible and invisible light sources (c and d) with results closely matching the ground truths (c.1 and d.1). It can insert virtual objects (e) with consistent changes of highlight and shadow and edit materials with color bleeding being correctly rendered image (f) and shading (f.1). It can also insert virtual lamps (g and h) and open a virtual window (i and j) to let sunlight (i.1 and j.1) shine into the room.

with accurate global interreflections, or morphing a room under fluorescent lights into one reflecting the sunrise through a window (Fig. 1).

Several recent works estimate *lighting* in indoor scenes [12,25,41,44], but achieving the above outcomes requires estimating and editing *light sources*. While both are highly ill-posed for single-image inputs, we posit that the latter presents fundamentally different and harder challenges for computer vision. First, it requires disentangling the individual contributions of both visible and invisible light sources, independent of the effects of geometry and material. Second, it requires reasoning about long-range effects such as interreflections, shadows and highlights, while also being precise about highly localized 3D shapes, spectra, directions and bandwidths of light sources, where minor errors can lead to global artifacts due to the above distant interactions. Third, it requires photo-realistic re-rendering of the scene despite only partial observations of geometry and material, while handling complex light transport. Figure 2 illustrates a few such challenges.

We solve the above challenges by bringing together a rich set of insights across physically-based vision and neural rendering. Given a single LDR image of an indoor scene, with predicted depth map and masks for visible lights, we propose to estimate *parametric models* of both visible and invisible light sources, in addition to per-pixel reflectance. Beyond a 3D location, our modeling accurately supports physical properties such as geometry, color, directionality and fall-off. Next, we design a *neural differentiable renderer* that judiciously uses classical methods and learned priors to synthesize high-quality images from predicted reflectance and light sources. We accurately model long-range light trans-

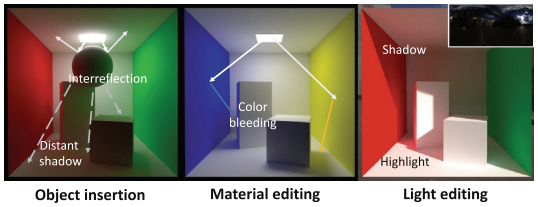

Fig. 2. Image editing must explicitly predict light sources to account for global effects such as distant shadows due to inserted objects, interreflections on far surfaces due to edited materials and light shafts by opening a window.

Table 1. Compared to prior works on inverse rendering, ours enables full scene relighting with global effects for inserted objects, edited materials or light sources. Also see Figs. 1 and 2.

	Input	Object insertion		Material editing		Light editing	
		Position	Non-local	Specular	Non-local	Lamp	Window
Auto, Karsch 14	Single	Any	✓	✗	✗	✓	✗
CGI, Li 18	Single	✗	✗	✗	✗	✗	✗
DeRenderNet, Zhu 21	Single	✗	✗	✗	✗	✗	✗
DeepPara, Gardner 19	Single	Any	✓	✗	✗	✗	✗
InvIndoor, Li 20	Single	Surface	✗	✓	✗	✗	✗
Lighthouse, Srinivasan 20	Stereo	Any	✗	✗	✗	✗	✗
FreeView, Philip 21	Multi.	✗	✗	✗	✗	✓	✗
Ours	Single	Any	✓	✓	✓	✓	✓

port through a physically-based Monte Carlo ray tracer with a learned shadow denoiser to render direct illumination, and an indirect illumination network to infer non-local interreflection. Our neural renderer injects the inductive bias of physical image formation in training, while allowing rendering and editing of global light transport from partial observations, as well as optimization to refine predictions.

Our parametric light source estimation and physically-based neural renderer allow intuitive editing of lamps and windows, with their global effects handled explicitly. In Fig. 1(c, d), we turn off each visible and invisible lamps. Beyond standard object insertion of prior works (e), we visualize inserted objects by "turning on" a new lamp (g, h) or "opening" a window with incoming sunlight (i, j). In each case, global effects such as highlights, shadows and interreflections are accurately created for the entire scene by the neural renderer, and are also properly handled when we edit materials of scene surfaces (f). In the accompanying video, we show that these editing effects are consistent as we move virtual objects and light sources, or gradually change materials. These abilities significantly surpass prior methods for intrinsic decomposition or inverse rendering. As summarized in Table 1 and Sect. 2, our method is the first to allow a broad range of single image scene relighting abilities in the form of inserting objects, changing complex materials and editing light sources, with consistent global interactions.

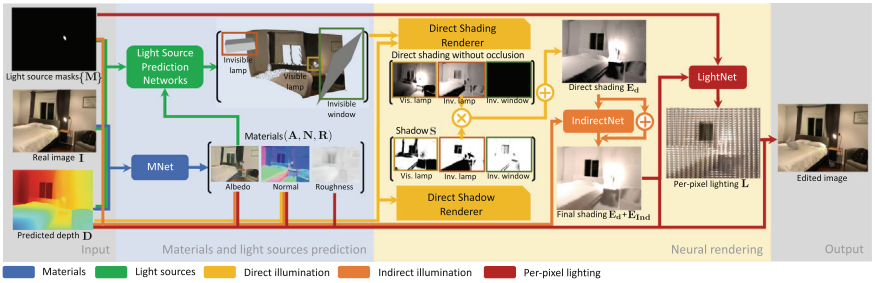

Fig. 3. Overview of our method. We start from a LDR RGB image, with depth map and visible light source masks estimated from the image or given as inputs. We first estimate per-pixel reflectance (albedo, normal, roughness) using a network (blue). Next, we estimate light sources (windows and lamps, visible and invisible) using four networks (green). To render the predictions back into an image, we use a neural renderer with three modules: direct shading, shadow (yellow), and indirect shading module (orange). The result is per-pixel shading (diffuse irradiance), which can be turned into per-pixel lighting (a grid of incoming radiance environment maps) using another network (red). (Color figure online)

2 Related Work

Inverse Rendering. Inverse rendering seeks to estimate factors of image formation (shape, materials and lighting) [30], which has traditionally required multiple images and controlled setups [7,9,14,45]. Several single-image works on material acquisition [22,26], or object-level shape and reflectance reconstruction use known [16,33] or semi-controlled lighting [27]. We consider a complex indoor scene under unknown illumination and jointly estimate its geometry, material and lighting from a single LDR image. Intrinsic decomposition [2–4,23,24,39] decomposes an image into Lambertian reflectance and diffuse shading. A recent work also predicts a shadow map [51]. Several deep learning methods estimate complex SVBRDFs and lighting [25,38]. But none of the above can estimate or edit light sources. We instead propose a novel physically-based 3D light source representation and neural rendering framework that estimates and edits individual light sources with distant shadows and global illumination being explicitly handled.

Lighting Estimation and Representation. Many single image approaches estimate lighting as a single environment map [10,11,21], which cannot express spatial variation of indoor illumination. Some recent works model spatial variations as per-pixel environment maps [1,13,25,50], or volumes [41,44]. However, these non-parametric representations can mainly be used for object insertion, while we estimate editable light sources with physically meaningful properties (position, geometry, direction, and intensity). Gardner et al. [12] predict a fixed number of spherical Gaussian lobes to approximate indoor light sources but do not handle light editing or its global effects. Zhang et al. recover geometry and

radiance of an empty room but cannot handle furniture inside [49]. Karsch et al. reconstruct geometry, reflectance and lighting but do not model windows and invisible scene contributions, require extensive user inputs [18] or face artifacts from imperfect heuristics or optimization [19]. In contrast, our physically-based neural renderer synthesizes photorealistic images with complex light transport, to enable relighting, light source insertion and removal from a single image.

Neural Rendering and Relighting. NeRF [31] and other volumetric neural rendering approaches have achieved photo-realistic outputs, but usually limited to view synthesis [29,31,48]. A few recent works [5,6,8,40,46] handle relighting, but use a per-object optimization from a large set of images. Philip et al. [35] demonstrate relighting for outdoor scenes but require multiple images. Concurrent to our work, Philip et al. [36] consider indoor relighting, but require a large number of high-resolution RAW images, cannot reconstruct directional sunlight and do not support material editing and object insertion with their neural renderer. As shown in Fig. 2 and Table 1, our modeling and neural rendering enable applications not possible for prior works, such as light source insertion/removal, virtual objects insertion and editing materials with non-local effects, from a single image.

3 Material and Light Source Prediction

Our overall framework is summarized in Fig. 3. In this section, we describe our novel, physically meaningful and editable reflectance and light source representations, while Sect. 4 describes our neural renderer that is differentiable with respect to light sources to facilitate training and editing of complex light transport. For per-pixel reflectance, we train a U-net similar to [25] to predict material parameters: diffuse albedo ($\mathbf{A}$), normal ($\mathbf{N}$) and roughness ($\mathbf{R}$), following the SVBRDF model of [17]. The inputs are a 240×320 LDR image ($\mathbf{I}$) and its corresponding depth map ($\mathbf{D}$), which in our case can be predicted by a state-of-the-art monocular depth prediction network [37]. We predict the normals directly, instead of computing them as the normalized gradient of depth to avoid artifacts and discontinuities. Thus, our prediction is given by $\{\mathbf{A}, \mathbf{N}, \mathbf{R}\} = \mathbf{MNet}(\mathbf{I}, \mathbf{D})$.

3.1 Light Source Representation

To enable high-quality indoor scene relighting, we need lighting representations that are editable, expressive enough for different types of lighting and realistic enough for convincing rendering of complex scenes. We model radiance and geometry of two types of common light sources with very different properties: (a) *windows* that can cover large areas and may induce strong directional sunlight, and (b) *lamps* that tend to be small and with more complex geometry.

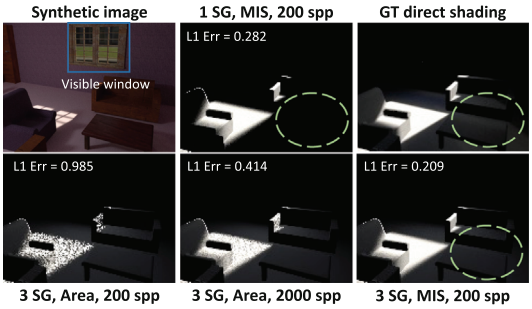

Fig. 4. Comparisons of direct shading rendered from different window representations with different sampling methods. We show that our 3 SGs models ambient lighting much better than a single SG, as shown in the green circle, and MIS sampling leads to much less noise compared to sampling window area uniformly. (Color figure online)

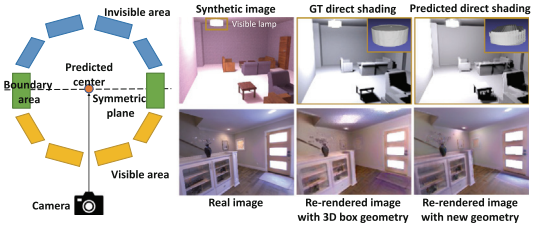

Fig. 5. A demonstration of our visible lamp geometry representation. Our representation for visible lamps is much less likely to cause highlight artifacts and wrong shadows compared to a standard 3D bounding box.

Radiance. The emitted radiance of lamps can be modeled by a standard Lambertian model, where every surface point with intensity $\mathbf{w}$ emits light uniformly. However, the radiance distribution of windows can be strongly directional due to sunlight coming through on a clear day, which is important for capturing realistic indoor lighting but often neglected by prior methods [36,41,42]. A recent work [44] models directional lighting with a single spherical Gaussian (SG), but as shown in Fig. 4, cannot recover ambient effects leading to sub-optimal rendering. Instead, we model the directional distribution of window radiance with 3 SGs corresponding to the sun, sky and ground. Each SG is defined by three parameters $\mathcal{G}_\mathbf{k} = (\mathbf{w}_\mathbf{k}, \lambda_\mathbf{k}, \mathbf{d}_\mathbf{k})$, for intensity, bandwidth and direction of lighting. For a ray in direction $\mathbf{l}$ that hits the window, its intensity is $\mathbf{L}_\mathcal{W}(\mathbf{l}) = \sum_\mathbf{k} \mathbf{w}_\mathbf{k} \exp\left(\lambda_\mathbf{k}(\mathbf{d}_\mathbf{k} \cdot \mathbf{l} - 1)\right)$, where $k \in \{\text{sun, sky, grnd}\}$. Figure 4 shows that our representation with multiple importance sampling leads to direct shading close to the ground-truth.

Geometry. Window geometry can be simply approximated by a rectangle $\{\mathbf{c}, \mathbf{x}, \mathbf{y}\}$, where $\mathbf{c}$ is the center and $\mathbf{x}, \mathbf{y}$ are the two axes. However, lamps present more diverse geometry. Naively representing a lamp with a 3D bounding box

$\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$ works for invisible lamps, but it often leads to artifacts for visible lamps, as the imperfect shape generates incorrect highlights. Therefore, we carefully design a new visible lamp representation shown in Fig. 5. We first identify the visible surface based on the depth $\mathbf{D}$ and lamp segmentation mask $\mathbf{M}_{\mathcal{L}}$, reconstruct the invisible surface by reflecting the visible surface with respect to the lamp center $\mathbf{c}$ and then add the boundary area. As shown in Fig. 5, our new representation can effectively constrain the lamp geometry and achieve realistic rendering without highlight artifacts for difficult real world examples. More details are in the supp.

3.2 Light Source Prediction

We use four neural networks to predict visible and invisible light sources for the lamp and window categories. For visible light sources, the inputs include extra instance segmentation masks. We can obtain the mask by either fine-tuning a Mask R-CNN [15] for our dataset, combined with a graph-cut based post processing to refine the boundaries, or manually draw the masks. While this is not our main focus, we include both qualitative and quantitative analysis in the supp. Let $\mathbf{M}_{\mathcal{W}}$ be a mask for a window and $\mathbf{M}_{\mathcal{L}}$ be a mask for a lamp. We have

$$\{\mathbf{c}, \mathbf{w}\} = \mathbf{VisLampNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}_{\mathcal{L}}),$$
$$\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathcal{G}_{\text{sun}}, \mathcal{G}_{\text{sky}}, \mathcal{G}_{\text{grnd}}\} = \mathbf{VisWinNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}_{\mathcal{W}}).$$

We assume one invisible lamp and one invisible window. These are deliberate simplifications: while invisible lights can contribute significant illumination, they are hard to infer using only indirect cues. We limit the expressivity of the representation to account for this ill-posedness and find it to be a good choice in practice[1]. When a scene has no invisible light source, their predicted intensities

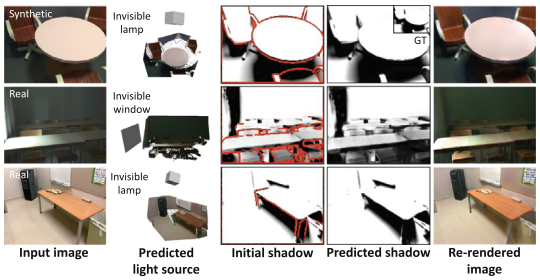

Fig. 6. Direct rendering shadows with ray tracing leads to boundary artifacts as shown in red color in the third column. Our trained depth-based shadow renderer achieves high-quality shadows for both real and synthetic scenes, with re-rendered images closely matching the inputs. (Color figure online)

[1] The real scene in Fig. 1 has 4 invisible lamps and the last real scene in Fig. 6 has 2. In both cases, we achieve reasonable approximation with one invisible lamp.

are close to zero, as shown in Fig. 3 and Fig. 8. To learn a better separation of visible and invisible light sources, we provide a mask $\mathbf{M} = \sum_{\mathcal{W}} \mathbf{M}_{\mathcal{W}} + \sum_{\mathcal{L}} \mathbf{M}_{\mathcal{L}}$ of all visible sources to the invisible light sources estimation networks:

$$\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathbf{z}\} = \mathbf{InvLampNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}),$$
$$\{\mathbf{c}, \mathbf{x}, \mathbf{y}, \mathcal{G}_{\text{sun}}, \mathcal{G}_{\text{sky}}, \mathcal{G}_{\text{grnd}}\} = \mathbf{InvWinNet}(\mathbf{I}, \mathbf{A}, \mathbf{D}, \mathbf{M}).$$

4 Neural Rendering Framework

To achieve photorealistic indoor light editing, we need a rendering framework that can handle complex light transport typical for indoor scenes, such as sharp directional lighting, hard and soft shadows and non-local interreflections. While existing differentiable path tracers can handle all these effects, they are computationally expensive. More importantly, they require the full reconstruction of reflectance and geometry of the entire scene, including its invisible parts.

To address these limitations, we introduce a neural rendering framework that combines the advantages of physically-based rendering and learning-based rendering. It works with our light source representations, does not require full scene reconstruction, achieves high performance, and is differentiable. Our framework, illustrated in Fig. 3 (right), has 4 modules: (1) a physically-based direct shading module that computes the direct irradiance from each light source through Monte Carlo sampling; (2) a hybrid shadow module that can render hard/soft shadows for each light source; (3) an indirect shading module that predicts non-local global illumination; (4) a per-pixel lighting module that predicts per-pixel environment map, which can be used to insert specular objects.

Our direct shading and shadows are computed based on ray tracing, while global illumination and per-pixel lighting are predicted by networks. The reason is that without full scene reconstruction, global illumination can only be computed heuristically (Fig. 7), which is suited for neural networks. Conversely, direct illumination and non-local shadowing can be efficiently computed by ray tracing, but remain tricky for neural methods.

Table 2. Shadow rendering error with or w/o network inpainting.

	Ray traced	Ours
L_2	0.011	0.005

4.1 Direct Shading Rendering Module

We use inspiration from physically-based rendering [34] to sample the surface of each light source and connect those samples to the scene points. Formally, let $\mathbf{p}$ be a shading point and $\mathbf{q}$ be a point uniformly sampled on the light surface,

with $\mathbf{p}\rightarrow\mathbf{q}$ the unit vector from $\mathbf{p}$ to $\mathbf{q}$. The direct shading $\mathbf{E_j}$ caused by light source $\mathbf{j}$ is computed as:

$$\mathbf{E_j}(\mathbf{p}) = \frac{\text{area}(\mathbf{j})}{N_\mathbf{j}} \sum_\mathbf{q} \frac{\mathbf{L_j}(\mathbf{q}\rightarrow\mathbf{p}) \max(\cos\theta_\mathbf{p} \cos\theta_\mathbf{q}, 0)}{||\mathbf{q}-\mathbf{p}||_2^2}, \tag{1}$$

where $\cos\theta_\mathbf{p} = \mathbf{p}\rightarrow\mathbf{q} \cdot \mathbf{N}(\mathbf{p})$, $\cos\theta_\mathbf{q} = \mathbf{q}\rightarrow\mathbf{p} \cdot \mathbf{N}(\mathbf{q})$ and $N_\mathbf{j}$ is the number of samples for light source $\mathbf{j}$. While our Monte Carlo estimation in (1) converges fast for lamps, it is not optimal for high-frequency directional sunlight coming through windows, since only when $\mathbf{q}\rightarrow\mathbf{p}$ aligns with the sun direction, will the $\mathbf{L}(\mathbf{q}\rightarrow\mathbf{p})$ return a significant contribution. To tackle this issue, with $\mathbf{Pr}(\mathbf{l})$ the probability of sampling direction $\mathbf{l}$ from $\mathcal{G}_{sun}$, we also generate samples according to the angular distribution of $\mathcal{G}_{\text{sun}}$:

$$\mathbf{E_j}(\mathbf{p}) = \sum_\mathbf{l} \frac{\mathbf{L_j}(\mathbf{l})\mathbf{I_j}(\mathbf{l}) \max(\cos\theta_\mathbf{p}, 0)}{N_\mathbf{j}\mathbf{Pr}(\mathbf{l})}, \tag{2}$$

where $\mathbf{I_j}(\mathbf{l})$ is an indicator function to detect if ray $\mathbf{l}$ starting from $\mathbf{p}$ can hit the window plane. Note that both (1) and (2) are unbiased but with different variances, which we combine with multiple importance sampling (MIS) [43]. Details are in the supp. Figure 4 compares the direct shading of a window, where we observe that our MIS method can render high-quality direct shading with much fewer samples, which makes training with rendering loss possible.

4.2 Depth-Based Hybrid Shadow Rendering Module

Recall that in the above shading computation, $\mathbf{E_j}, j \in \{\mathcal{W}\} \cup \{\mathcal{L}\}$ does not consider visibility and therefore cannot handle shadows. We could check visibility by ray tracing during the Monte Carlo sampling above, but this causes artifacts due to incomplete geometry, as shown in Fig. 6. We instead design a depth-based shadow rendering framework that combines Monte Carlo ray tracing with learning-based inpainting and denoising. Our shadow modules are not differentiable, as this is not necessary for our application: we train our network on a synthetic dataset, which provides the ground truth direct shading without the shadow effects, so back-propagation of error through the shadow renderer is not necessary.

Our approach first creates a mesh from the depth map, and then uses a GPU-based ray tracer to cast shadow rays from surfaces to light sources. To address the boundary artifacts, we first modify the renderer to detect the occlusion boundaries, then train a CNN to fill in the shadow at these regions. This hybrid approach outperforms both pure ray tracing and a CNN trained to clean up the entire ray traced shadow image. Formally, let $\mathbf{S^{Init}}$ be the initial shadow image rendered from depth map $\mathbf{D}$ and let $\mathbf{M^S}$ be the mask for occlusion boundaries.

$$\mathbf{S} = \mathbf{M^S} \cdot \mathbf{DShdNet}(\mathbf{S^{Init}}, \mathbf{D}, \mathbf{N}) + (\mathbf{1} - \mathbf{M^S}) \cdot \mathbf{S^{Init}}. \tag{3}$$

The total direct shading from all sources is $\mathbf{E_d} = \sum_\mathbf{j} \mathbf{E_j}\mathbf{S_j}$. As seen in Fig. 1, 6 and 7, our framework can render higher quality soft and hard shadows that are

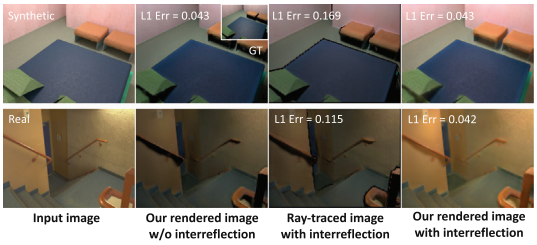

Fig. 7. Our neural renderer models both direct and indirect illumination accurately, while a ray tracer using only single-view predictions cannot model indirect illumination and has artifacts near occlusion boundaries.

closer to the ground-truths compared to a standard ray tracer. Table 2 shows that our CNN reduces the shadow error by more than 50%.

4.3 Indirect Shading Prediction

To render indirect illumination with a ray tracer, we would need full reconstruction of scene reflectance and geometry, which is infeasible from a single image. Instead, we train a 2D CNN to predict indirect shading in screen space. A similar idea was adopted by a recent work [47]. We use a network with large receptive field covering the entire image to model non-local inter-reflections. Our indirect shading is $\mathbf{E_{Ind}} = \mathbf{IndirectNet}(\mathbf{E_d}, \mathbf{D}, \mathbf{N}, \mathbf{A})$, which is added to the direct shading for the final shading prediction. In Fig. 7, we compare the indirect illumination rendered by our network and by a ray tracer using an incomplete textured mesh built from depth map and reflectance map predicted from a single image. Quantitative and qualitative results on real and synthetic examples show that our neural rendering layer renders both direct and indirect illumination accurately, while a ray tracer cannot handle indirect illumination with partial geometry and reflectance, leading to a darker image with similar intensity as the one with direct illumination only.

4.4 Predicting Lighting from Shading

The above framework cannot yet handle specular reflectance, which motivates us to add another network to infer spatially varying per-pixel lighting $\mathbf{L}$, taking the above shading (irradiance) $\mathbf{E}$ as input. We follow [25] to predict a grid of environment maps. We use a similar network architecture but replace the input image $\mathbf{I}$ with the shading $\mathbf{E}$ so that the predicted *local* lighting is a function of our lighting representation: $\mathbf{L} = \mathbf{LightNet}(\mathbf{E}, \mathbf{M}, \mathbf{A}, \mathbf{N}, \mathbf{R}, \mathbf{D})$. The predicted $\mathbf{L}$ can be used to render specular materials, shown in Fig. 11 and Fig. 12 in Sect. 5.

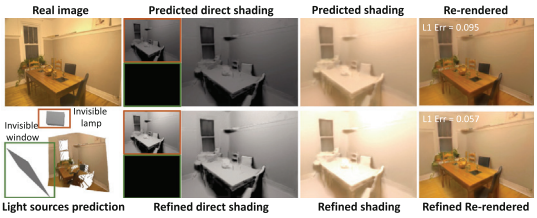

Fig. 8. Comparisons of light source prediction and rendering before and after the optimization on a real scene. Our neural renderer allows using the rendering loss to learn and refine light source intensity and direction

4.5 Implementation Details

Dataset. We train on OpenRooms [28] – a large-scale synthetic indoor dataset for inverse rendering – which is unique among currently available datasets in providing ground truths for all our outputs, such as light source geometry, per-light source shadings (with and without occlusion) and per-light source shadows. Thus, it allows to train each module separately, significantly simplifying training.

Optimized Light Source Parameters. We augment the OpenRooms dataset with optimized light source parameters $\{\mathcal{G}_{\text{sun}}, \mathcal{G}_{\text{sky}}, \mathcal{G}_{\text{grd}}\}$ for windows, leading to sharper and more interpretable predictions. To compute those, we minimize the L_1 difference between the rendered direct shading without occlusion $\mathbf{E_j}$, $j \in \{\mathcal{W}\}$ and its corresponding ground truth, through our differentiable Monte Carlo rendering module (Sect. 4.1). More details are in the supp. The optimized direct shading is seen in Fig. 4 to closely match the ground truth.

Losses. We use L_2 loss to train **MNet**. The loss function for light source prediction is the sum of a rendering loss ($\mathbf{Loss_{ren}}$), a geometry loss ($\mathbf{Loss_{geo}}$), and a light source loss ($\mathbf{Loss_{src}}$). For $\mathbf{Loss_{ren}}$, we define it to be the L_1 distance between the rendered direct shading $\mathbf{E_j}$ and its ground-truth, without shadows applied. For $\mathbf{Loss_{geo}}$, we uniformly sample points $\{\mathbf{q}\}$ from the ground-truth and predicted light source geometry to compute their RMSE Chamfer distances and add an L_1 loss for its surface area to encourage sharper lighting. For $\mathbf{Loss_{src}}$, we use L_2 loss for direction $\mathbf{d}$, $\log L_2$ loss for intensity $\mathbf{w}$ and bandwidth λ. To train the shadow network, we use scale-invariant gradient loss proposed in [32] and find that it leads to many fewer artifacts compared to a simple L_2 loss. We supervise indirect shading with L_1 loss and per-pixel lighting with rendering loss and $\log L_2$ loss similar to [25]. More details are in the supp.

Training and Inference. We use Adam [20] with learning rate 10^{-4} and β (0.9, 0.999). We first train the **MNet** and then use its predictions as inputs to train **InvLampNet**, **InvWinNet**, **VisLampNet** and **VisWinNet** separately. We also train rendering modules independently by providing them with ground-truth $\mathbf{E_d}$ and $\mathbf{S}$. The typical inference time is less than 3s. More details are in the supp.

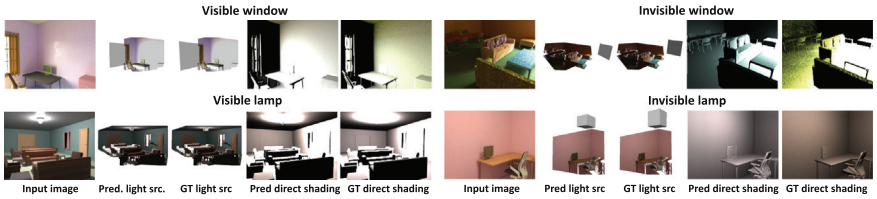

Fig. 9. Light source prediction on our synthetic dataset for four types of light sources. We visualize light source geometry and direct shading $\mathbf{E_j}$ without occlusion. Our method recovers both geometry and radiance of four types of light sources reasonably well.

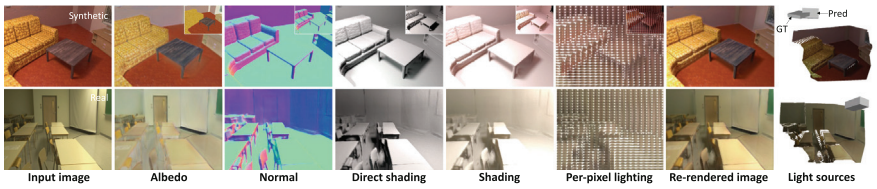

Fig. 10. Our reflectance, lighting and rendering results on a synthetic and a real example. Ground truths for the synthetic example are shown in the insets. We observe that even for invisible light sources, our framework accurately reconstructs their geometry and radiance, which enables realistic rendering of shadings, shadows, interreflections and per-pixel lighting and final images.

Refinement. While so far our framework can achieve high-quality light source prediction and indoor lighting editing in many cases, our differentiable neural renderer enables us to further refine the light source parameters by minimizing the rendering loss between the rendered and the input image. Figure 8 shows an example where we correct the intensity of an invisible lamp with our rendering loss-based refinement. Note that as this is an extremely ill-posed problem, good initialization from our network predictions is essential for the refinement to achieve good results. More discussions are in the supp. We only apply the refinement to real images shown in the paper, not to the synthetic images.

5 Experiments

We present light source estimation and neural rendering results on real and synthetic data, as well as various scene editing applications, especially light editing, on real data. For synthetic data, we test both ground-truth and predicted depths from DPT [37] w/o fine-tuning and use ground truth light source masks. For real data, we generate all depth predictions using DPT [37] and manually draw light source masks. While not being our main focus, we also evaluate a Mask RCNN [15] for light source detection in the supp.

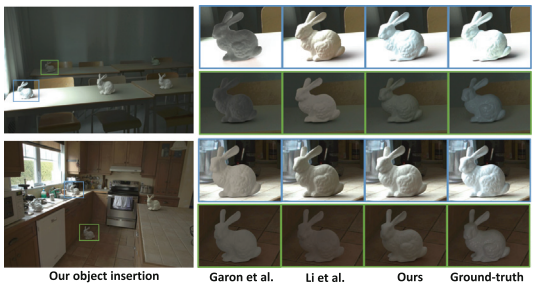

Fig. 11. We achieve similar quality as prior state-of-the-art on Garon et al. [13] dataset for object insertion. Our method accurately reconstructs the complex lighting from windows to render more realistic highlights and shadows. See Fig. 12 for other editing tasks not possible for prior works.

Table 3. Light source prediction on OpenRooms with ground truth and predicted depth. We report RMSE chamfer loss and L_1 error of direct shading w/o shadows $\mathbf{E_j}$.

	Geometry		Rendering	
	Cham $(\mathbf{q_j}, \bar{\mathbf{q}}_\mathbf{j})$		$\mathbf{E_j}$	
	Gt.	Pred.	Gt.	Pred.
Vis. lamp	0.279	1.15	0.317	0.557
Vis. window	0.415	1.14	0.849	0.952
Inv. lamp	0.712	0.988	0.289	0.357
Inv. window	3.50	3.71	0.312	0.328

Table 4. Quantitative errors for our neural rendering framework on OpenRooms with ground-truth and predicted depth. We report L_1 loss for the sum of direct shading with shadows $\mathbf{E_d}$ and shading with global illumination $\mathbf{E}$. We report log L_2 loss for per-pixel lighting $\mathbf{L}$.

Direct shading $\mathbf{E_d}$		Shading $\mathbf{E}$		Perpix. envmap $\mathbf{L}$	
Gt.	Pred.	Gt.	Pred.	Gt.	Pred.
0.283	0.325	0.336	0.391	0.090	0.105

Light Source Predictions and Neural Rendering. Figure 9 shows qualitative results on synthetic images with ground truth depth. Qualitative synthetic results with predicted depth are in the supp. We observe that our method can recover both the geometry and radiance for all 4 types of light sources reasonably well, which enables us to render their direct shading quite close to the ground-truths. The major errors are global shifts of colors and intensities, while the locations of highlights are usually correct. This is reasonable given the ambiguities between materials and lighting. Table 3 reports the quantitative errors with both ground truth and predicted. The errors for windows are larger than those of lamps, since the outdoor lighting coming through windows is much more complicated compared to area lighting. In addition, the direct shading errors for invisible light sources are lower. This is because their overall contributions are usually lower since many of them are far away from the camera location. We observe that our method also achieves comparable rendering errors even with predicted depth, suggesting that it can generalize well to inaccurate geometry.

Table 5. User study on Garon et al. dataset.

Gardner et al. [11]	Garon et al. [13]	Li et al. [28]
72.4%	69.2%	52.0%

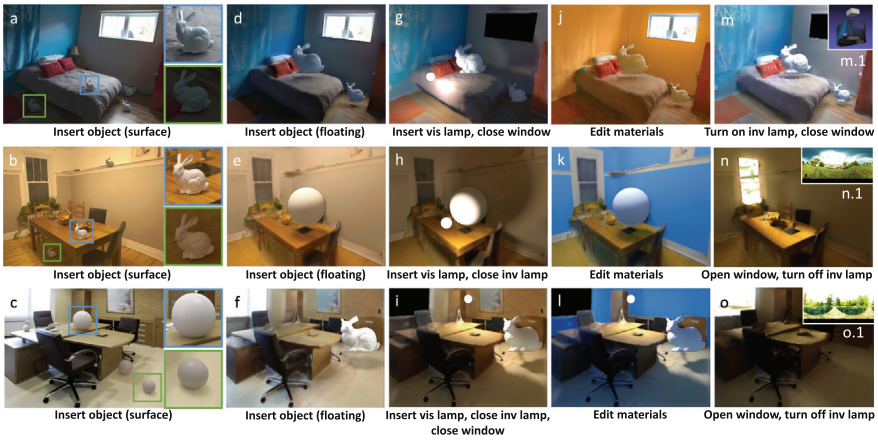

Fig. 12. Various editing applications demonstrated on 3 real examples. In addition to high-quality object insertion (a, b and c), our framework allows editing geometry, material and lighting of indoor scenes, with consistent non-local effects. This includes distant shadows projected to the bed, table and floor (d, e, f and i) or to the entire room when the object blocks the light source (g and h), changing color of walls that causes non-local color bleeding (j, k and l) and adding virtual light sources into the scene (g, h, i, l, m, n, o), such as turning on a lamp or opening a virtual window. (Color figure online)

Figure 10 shows our neural rendering results on a synthetic and a real example. Quantitative results are summarized in Table 4. For the synthetic example, our shadow prediction network combined with Monte-Carlo ray tracing can render distant shadows from a single depth map without boundary artifacts. Our indirect shading prediction network models non-local interreflections from only single-view reconstruction of geometry and materials. All the modules combined together lead to accurate reconstruction of shading and per-pixel lighting. For the real example, even though we do not have ground truths, we observe that the light source position, the highlight in the direct shading and shadows are all visually consistent. The re-rendered image closely matches the input, which further demonstrates that our framework can generalize well to real examples.

Comparisons with Prior Works. We reiterate that our method enables applications (e.g. light source editing) that are not possible with any prior work. While this makes direct comparisons challenging, we compare on a subset of tasks like object insertion that prior works support. We use Garon et al. dataset [13] for comparison, which is a widely-used, real dataset for spatially-varying lighting

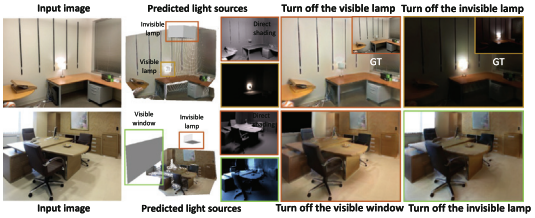

Fig. 13. Our accurate reconstruction of visible/invisible light sources allows separating their contributions and turn them on and off. Our results closely match the ground-truth insets.

evaluation. We conduct a user study by requiring 200 users to compare our results with prior results and report the percentage of users who believes ours are better. Even though we are solving a harder problem, both qualitative and quantitative results in Fig. 11 and Table 5 show that our method achieves performance comparable to the prior state-of-the-arts which only handle local editing of the scene. Our per-pixel lighting prediction can be used to render specular objects realistically, with highlights, shadows and spatial consistency being correctly modeled. Specifically, our window representation and MIS based rendering layer can better handle high-frequency, complex sunlight, leading to rendering results closer to the ground truths, as presented in Fig. 11.

Novel Scene Editing Applications. In addition to object insertion (a, b, c) with realistic highlights and shadows, the true advantage of our framework is its ability to handle non-local effects in novel scene editing applications, which is only made possible by our accurate reconstruction of indoor light sources and high-quality neural rendering framework. These non-local effects include distant shadows and highlights, which is shown in (d, e, f) of Fig. 12 where the inserted virtual objects block the light coming from the visible window or the invisible lamp. This is further demonstrated in (g, h, i), where the inserted virtual lamp causes highlights on the nearby geometry and shadows that cover the whole wall behind the virtual bunny and sphere. Moreover, our framework can model non-local interreflection accurately. As shown in (j, k, l), as we change the color of walls to orange and blue, our indirect shading network paints the inserted white objects with correct color bleeding. In (m, n, o), we demonstrate our framework's ability to turn on an invisible lamp or open a virtual window. In n, o, we use the 3 SG approximation of the environment map shown in n.1 and o.1 respectively. Our representation combined with our neural renderer can render realistic directional sunlight. Our accurate reconstruction of indoor light sources further allows us to separate their contributions. As shown in both Fig. 1 and 13, our framework allows turning off visible and invisible, lamps or windows in the scene, with changed appearance similar to the ground-truth insets[2].

[2] The second example is from the internet so we do not have its ground truth.

Please see *supplementary material* for ablation studies, error distributions, failure cases, limitations and a video illustrating consistent scene editing effects as we move virtual objects and light sources, or gradually change the wall color.

6 Conclusions

We presented a method that enables full indoor scene relighting and other editing operations from a single LDR image with its predicted depth and light source segmentation masks. The first key innovation is our lighting representation; we estimate multiple global 3D parametric lights (lamps and windows), both visible and invisible. The second is our hybrid neural renderer, capable of producing high-quality images from our representations using a combination of Monte Carlo and neural techniques. We show that this careful combination can for the first time handle challenging scene editing applications including object insertion, material editing, light source insertion and editing, with realistic global effects.

Acknowledgment. We thank NSF CAREER 1751365, 2110409, 1703957, CHASE-CI, ONR N000142012529, N000141912293, a Google Award, gifts from Adobe, Ron L. Graham Chair, UCSD Center for Visual Computing and Qualcomm Fellowship.

References

1. Barron, J.T., Malik, J.: Shape, illumination, and reflectance from shading. PAMI **37**(8), 1670–1687 (2015)
2. Barrow, H.G., Tenenbaum, J.M.: Recovering intrinsic scene characteristics from images. Comput. Vis. Syst. 3–26 (1978)
3. Bell, S., Bala, K., Snavely, N.: Intrinsic images in the wild. ACM Trans. Graph. (TOG) **33**(4), 159 (2014)
4. Bi, S., Han, X., Yu, Y.: An l 1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. ACM Trans. Graph. (TOG) **34**(4), 1–12 (2015)
5. Bi, S., et al.: Neural reflectance fields for appearance acquisition. arXiv preprint arXiv:2008.03824 (2020)
6. Bi, S., et al.: Deep reflectance volumes: relightable reconstructions from multi-view photometric images. arXiv preprint arXiv:2007.09892 (2020)
7. Bi, S., Xu, Z., Sunkavalli, K., Kriegman, D., Ramamoorthi, R.: Deep 3D capture: geometry and reflectance from sparse multi-view images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5960–5969 (2020)
8. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: NeRD: neural reflectance decomposition from image collections. arXiv preprint arXiv:2012.03918 (2020)
9. Chandraker, M.: On shape and material recovery from motion. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 202–217. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_14
10. Debevec, P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: SIGGRAPH, vol. 98, pp. 189–198 (1998)

11. Gardner, M.A., et al.: Learning to predict indoor illumination from a single image. ACM Trans. Graph. **9**(4) (2017)
12. Gardner, M.A., Hold-Geoffroy, Y., Sunkavalli, K., Gagne, C., Lalonde, J.F.: Deep parametric indoor lighting estimation. In: ICCV (2019)
13. Garon, M., Sunkavalli, K., Hadap, S., Carr, N., Lalonde, J.F.: Fast spatially-varying indoor lighting estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6908–6917 (2019)
14. Goldman, D.B., Curless, B., Hertzmann, A., Seitz, S.M.: Shape and spatially-varying BRDFs from photometric stereo. PAMI **32**(6), 1060–1071 (2010)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
16. Johnson, M.K., Adelson, E.H.: Shape estimation in natural illumination. In: CVPR (2011)
17. Karis, B., Games, E.: Real shading in unreal engine 4. In: Proceedings of Physically Based Shading Theory Practice
18. Karsch, K., Hedau, V., Forsyth, D., Hoiem, D.: Rendering synthetic objects into legacy photographs. ACM Trans. Graph. **30**(6), 1 (2011)
19. Karsch, K., et al.: Automatic scene inference for 3d object compositing. ACM Trans. Graph. **33**, 32:1–32:15 (2014)
20. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. LeGendre, C., et al.: DeepLight: learning illumination for unconstrained mobile mixed reality. In: CVPR, pp. 5918–5928 (2019)
22. Li, X., Dong, Y., Peers, P., Tong, X.: Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. ACM Trans. Graph. **36**(4), 1–11 (2017)
23. Li, Z., Snavely, N.: CGIntrinsics: better intrinsic image decomposition through physically-based rendering. In: ECCV, pp. 371–387 (2018)
24. Li, Z., Snavely, N.: Learning intrinsic image decomposition from watching the world. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9039–9048 (2018)
25. 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 (2020)
26. Li, Z., Sunkavalli, K., Chandraker, M.: Materials for masses: SVBRDF acquisition with a single mobile phone image. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 74–90. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_5
27. Li, Z., Xu, Z., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. In: SIGGRAPH Asia, p. 269. ACM (2018)
28. Li, Z., et al.: OpenRooms: an end-to-end open framework for photorealistic indoor scene datasets. In: CVPR (2021)
29. Liu, L., Gu, J., Lin, K.Z., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. arXiv preprint arXiv:2007.11571 (2020)
30. Marschner, S.: Inverse rendering for computer graphics (1998)
31. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 405–421. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_24

32. Niklaus, S., Mai, L., Yang, J., Liu, F.: 3D ken burns effect from a single image. ACM Trans. Graph. (TOG) **38**(6), 1–15 (2019)
33. Oxholm, G., Nishino, K.: Shape and reflectance from natural illumination. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7572, pp. 528–541. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33718-5_38
34. Pharr, M., Jakob, W., Humphreys, G.: Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann (2016)
35. Philip, J., Gharbi, M., Zhou, T., Efros, A.A., Drettakis, G.: Multi-view relighting using a geometry-aware network. ACM Trans. Graph. (TOG) **38**(4), 1–14 (2019)
36. Philip, J., Morgenthaler, S., Gharbi, M., Drettakis, G.: Free-viewpoint indoor neural relighting from multi-view stereo. ACM Trans. Graph. **40**, 1–18 (2021)
37. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV, pp. 12179–12188 (2021)
38. Sengupta, S., Gu, J., Kim, K., Liu, G., Jacobs, D.W., Kautz, J.: Neural inverse rendering of an indoor scene from a single image. arXiv preprint arXiv:1901.02453 (2019)
39. Shen, J., Yang, X., Jia, Y., Li, X.: Intrinsic images using optimization. In: CVPR 2011, pp. 3481–3487. IEEE (2011)
40. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: NeRV: neural reflectance and visibility fields for relighting and view synthesis. arXiv preprint arXiv:2012.03927 (2020)
41. Srinivasan, P.P., Mildenhall, B., Tancik, M., Barron, J.T., Tucker, R., Snavely, N.: Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8080–8089 (2020)
42. Straub, J., et al.: The Replica dataset: a digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
43. Veach, E.: Robust Monte Carlo methods for light transport simulation, vol. 1610. Stanford University Ph.D. thesis (1997)
44. Wang, Z., Philion, J., Fidler, S., Kautz, J.: Learning indoor inverse rendering with 3D spatially-varying lighting. In: ICCV (2021)
45. Xia, R., Dong, Y., Peers, P., Tong, X.: Recovering shape and spatially-varying surface reflectance under unknown illumination. ACM Trans. Graph. **35**(6), 187 (2016)
46. Xiang, F., Xu, Z., Hašan, M., Hold-Geoffroy, Y., Sunkavalli, K., Su, H.: NeuTex: neural texture mapping for volumetric neural rendering. arXiv preprint arXiv:2103.00762 (2021)
47. Xin, H., Zheng, S., Xu, K., Yan, L.Q.: Lightweight bilateral convolutional neural networks for interactive single-bounce diffuse indirect illumination. IEEE Ann. Hist. Comput. (01), 1 (2020)
48. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: neural radiance fields from one or few images. arXiv preprint arXiv:2012.02190 (2020)
49. Zhang, E., Cohen, M.F., Curless, B.: Emptying, refurnishing, and relighting indoor spaces. ACM Trans. Graph. (TOG) **35**(6), 1–14 (2016)
50. Zhou, H., Yu, X., Jacobs, D.W.: GLoSH: global-local spherical harmonics for intrinsic image decomposition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7820–7829 (2019)
51. Zhu, Y., Tang, J., Li, S., Shi, B.: DeRenderNet: intrinsic image decomposition of urban scenes with shape-(in) dependent shading rendering. In: 2021 IEEE International Conference on Computational Photography (ICCP), pp. 1–11. IEEE (2021)