Deep graph learning for spatially-varying indoor lighting prediction

Jiayang BAI, Jie GUO*, Chenchen WANG, Zhenyu CHEN, Zhen HE, Shan YANG, Piaopiao YU, Yan ZHANG & Yanwen GUO*

State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210023, China

Received 7 April 2022/Accepted 4 July 2022/Published online 18 January 2023

Abstract  Lighting prediction from a single image is becoming increasingly important in many vision and augmented reality (AR) applications in which shading and shadow consistency between virtual and real objects should be guaranteed. However, this is a notoriously ill-posed problem, especially for indoor scenarios, because of the complexity of indoor luminaires and the limited information involved in 2D images. In this paper, we propose a graph learning-based framework for indoor lighting estimation. The core is a new lighting model (DSGLight) based on depth-augmented spherical Gaussians (SGs) and a graph convolutional network (GCN) that infers the new lighting representation from a single low dynamic range (LDR) image of limited field-of-view. Our lighting model builds 128 evenly distributed SGs over the indoor panorama, where each SG encodes the lighting and the depth around that node. The proposed GCN then learns the mapping from the input image to DSGLight. Compared with existing lighting models, our DSGLight encodes both direct lighting and indirect environmental lighting more faithfully and compactly. It also makes network training and inference more stable. The estimated depth distribution enables temporally stable shading and shadows under spatially-varying lighting. Through thorough experiments, we show that our method obviously outperforms existing methods both qualitatively and quantitatively.

Keywords  lighting, graph learning, augmented reality, spherical Gaussians, rendering

Citation  Bai J Y, Guo J, Wang C C, et al. Deep graph learning for spatially-varying indoor lighting prediction. Sci China Inf Sci, 2023, 66(3): 132106, https://doi.org/10.1007/s11432-022-3576-9

1 Introduction

Predicting indoor illumination from a single image is a challenging but meaningful task [1]. It is a fundamental step for many applications, ranging from indoor scene understanding to augmented reality (AR). However, this problem is highly ill-posed due to the following reasons. First, indoor illumination stems from a wide range of different luminaires including spot lights, large area lights, and indirect lighting. This is quite different from outdoor lighting which is dominated by the sun [2–6]. Second, a single image captured by an ordinary camera has a limited field-of-view (FoV) that may fail to provide sufficient cues of indoor light sources. In fact, many indoor light sources have a local impact and yield spatially-varying lighting effects. This makes them hard to be identified from a single FoV-limited image. Moreover, strong lighting-reflectance ambiguities exist in indoor scenes since different combinations of lighting and surface reflectance may produce the same pixel intensities.

Currently, this problem is addressed in two ways. One line of work assumes the availability of some prior knowledge about the scene, e.g., scene geometries [7–9] obtained by depth sensors or multi-view inputs. Prior knowledge is effective for overcoming the ill-posedness of the problem but will limit the applicability of the methods. A recent trend is to leverage deep learning-based solutions accompanied with large training datasets to infer lighting from a single, casually captured image [10–16]. The lighting models used in these solutions can be divided into two categories. In the first category, the indoor lighting of surrounding environment is stored in a densely sampled environment map [11], i.e., image-based lighting (IBL). Unfortunately, inferring a full environment map directly is very difficult due to the
high dimensionality of the representation. Alternatively, the second category leverages analytical models with a small number of parameters to fit real-world illumination. For instance, indoor illumination can be approximated by a set of spherical harmonics (SH) \[12, 17-19\] or spherical Gaussians (SGs) \[10, 15, 20-22\]. Generally, SH works well for low-frequency lighting, but is plagued with ringing artifacts as the SH order increases. Gardner et al. \[10\] proposed using 2–5 SGs to model high-frequency light sources in an indoor scene. Although it outperforms SH-based methods, regressing the positions of floating SGs is unstable for deep learning. Consequently, this method achieves degraded performance as the number of SGs increases.

To retain the advantages of SGs in compactly encoding high-frequency lighting and enable stable training, we introduce DSGLight, a novel SG-based lighting model augmented with depth values. In this model, \(N\) SGs are assumed to have fixed positions and are evenly distributed over a unit sphere. We also fix the bandwidth of each SG such that only its color and depth values vary. In this case, predicting indoor lighting boils down to estimating \(N\) RGBD values, which is more stable and robust than regressing several free SGs on the sphere. The augmented depth values allow us to generate spatially-varying shading and shadows with respect to the insert location.

Considering the non-Euclidean nature of DSGLight, we design a graph convolution network (GCN) for feature extraction and lighting prediction. GCNs have gained impressive successes in many vision tasks, but are less exploited in lighting prediction. In our network, we first extract features from an input image with vanilla convolutional layers. Then, these features are connected to GCN features with several fully-connected layers. After information exchanging across neighboring nodes, we finally obtain the color and depth for each SG. We demonstrate through extensive experiments that our method outperforms existing methods both qualitatively and quantitatively. The state-of-the-art performance on indoor lighting prediction allows us to achieve more realistic shading results when inserting virtual objects into real indoor scenes (see Figure 1).

In summary, our main contributions in this paper are as follows:

- A depth-augmented indoor lighting model, i.e., DSGLight, to compactly and faithfully encode spatially-varying lighting in an indoor scene.
- A dedicated graph convolutional network for predicting DSGLight from a single image.
- A new dataset with supervision of DSGLight parameters that is constructed to train our network.

2 Related work

2.1 Indoor lighting prediction

Inferring the illumination from 2D images is a classic problem that has been extensively studied in the past decades. Early progress on illumination estimation relied on prior knowledge of scene geometry. For instance, Barron and Malik \[7\] adopted depth sensors to capture geometric information, Wu et al. \[23\] reconstructed geometry with multi-view stereo, and Karsch et al. \[24\] required users to annotate the geometry. Some other studies manage to predict illumination from purely 2D image(s). Khan et al. \[25\] presented a method to flip the given image to approximate the rest of the whole panorama, which is out of view. Unfortunately, this only works well in some special cases.

While deep learning is significantly successful in many graphical and vision tasks, some learning-based methods are proposed to estimate illumination. Wei et al. \[26\] decomposed the RGBD input into albedo, diffuse shading and specular shading maps, and thus they can integrate physically-based principles of inverse rendering together with deep learning to estimate illumination. Hold-Geoffroy et al. \[3\] presented
an end-to-end approach that leveraged convolutional neural network (CNN) to predict outdoor illumination represented by a low-dimensional analytical model, significantly outperforming previous traditional methods. Gardner et al. [11] also adopted an end-to-end CNN to regress a non-parametric environment map to represent accurate illumination. Song et al. [13] proposed “Neural Illumination” that can predict indoor lighting at a specific locale. Another recent study [14] leveraged a multi-scale volumetric lighting representation to estimate spatially-coherent illumination from images.

Close to our work, several methods also leverage multiple SGs to represent indoor lighting [10, 15]. However, these methods only capture dominated lights and do not account for spatially-varying environmental lights, resulting in fewer details for rendering specular objects and inconsistent shadows. Moreover, these methods usually leverage CNNs to establish the mapping between input images and SGs. However, CNNs, with a limited receptive field, fail to explore the long-range correlations between SGs.

2.2 Graph convolution networks

To deal with non-Euclidean data like graphs, GCNs provide well-suited solutions which can be divided into two types: spatial-based [27–29] and spectral-based [30–32]. Spatial-based methods used in many studies are mainly based on the spatial relations of vertices in graphs and they apply spectral convolutions on graph using ideas from graph signal processing [33]. As transforming the signal to spectral domains can be expensive, many recent studies [30, 32] approximate the spectral convolutions using Chebyshev polynomials. Similarly, our network also uses Chebyshev polynomials for spectral convolutions in GCN layers.

GCNs are becoming increasingly popular as graphs are widely used in many applications. For example, GCNs can be applied for natural language processing [34, 35]. It is also used to model human joints for action recognition in video [36] and predict semantic relations between object pairs [37–39]. Although GCNs are widely used, as far as we know, they are less studied in illumination estimation problem. In this paper, we use GCNs to extract features from indoor illumination encoded in a non-Euclidean form and establish the connection between 2D images and an SG-based lighting model.

3 DSGLight

3.1 Representation

DSGLight is a depth-augmented parametric lighting model that leverages a set of discrete SG functions to represent the spatially-varying indoor illumination, as illustrated in Figure 2. We define DSGLight by $\mathcal{M} = \{m_i\}_{i = 1, \ldots, N}$ which contains $N$ vertices. Each vertex in DSGLight is associated with a spherical Gaussian [21] defined as follows:

$$G(v; a, \lambda, \mu) = ae^{(v - \mu)^T},$$

where $\mu$ is the lobe axis (the centering direction), $\lambda \in (0, +\infty)$ is the lobe sharpness (the bandwidth), and $a$ is the lobe amplitude ($a \in \mathbb{R}^4$ for RGBD information). The direction $v$ denotes the spherical parameter of the resulting function. Given $m_i = \{a_i, \lambda_i, \mu_i\}$ for $i$th vertex, the overall indoor illumination encoded in a given panorama is compressed into an arbitrary spherical function $f(v)$ as follows:

$$f(v) = \sum_{i}^{N} G(v; m_i).$$

The corresponding depth panorama is treated in a similar way. Currently, we set $N$ to 128.

To further reduce the number of parameters, we fix the position $\mu$ and the bandwidth $\lambda$ of each SG. Specifically, 128 nodes are placed evenly over a unit sphere, and they share the same bandwidth. This allows $\mu$ and $v$ to be preset for our model. The distribution of these nodes is shown in Figure 2. Besides being a more compact representation, such a design stabilizes network training and inferences. As pointed out in [10], regressing several floating SGs with floating positions and unconstrained bandwidth is unstable, especially for many SGs. As a result, performance decreases from 3 to 5 SGs. With fixed position and bandwidth, inferring DSGLight boils down to predicting $N$ RGBD values, which is easier and more stable than inferring floating SGs.
3.2 Choice of the SG bandwidth

To make \( N \) SGs cover the whole panorama as much as possible and avoid overlapping, we set the bandwidth parameter \( \lambda \) as follows:

\[
\lambda = \frac{\ln 0.6}{\cos(\arctan(2/\sqrt{N})) - 1}. \tag{3}
\]

Assuming that the SG weight is \( w \), this bandwidth ensures that the contribution of each SG at the midpoint of the line segment of two adjacent SGs is 0.6\( w \). In this way, most DSGLight will cover the whole panorama with relatively small overlaps.

Figure 3 shows synthesized RGB panoramas with an increasing bandwidth. As seen, when the bandwidth is small, there will be many gaps and mosaics between adjacent SGs. Conversely, a very large bandwidth will lead to too many overlapped regions, making the resulting DSGLight overly-smooth. Generally, a bandwidth which contributes 0.6\( w \) at the midpoint of the line segment of two adjacent SGs achieves a good trade-off.

3.3 Choice of the number of SG

We show synthesized RGB panoramas represented with different numbers of SGs in Figure 4. We observe that the synthesized panoramas only maintain the main lighting with 32 or 64 SGs. When the number of SGs increases, the synthesized panoramas are capable to represent more details about the ambient illumination, which leads to a higher peak signal-to-noise ratio (PSNR). However, the increasing in PSNR is not significant when we use more than 128 SGs. Meanwhile, more SGs require more memory and computation power in training and inference. Considering the trade-off between performance and accuracy, we set the number of SGs to 128 in our experiments.

3.4 Comparison and analysis

Previous studies have proposed many parametric methods to reduce the number of parameters in representing panoramic lighting, such as SH, SG, and IBL. Here, we compare these representations with our DSGLight. One example comparing these representations is provided in Figure 5. Some information is summarized and compared in Table 1.
Among these methods, IBL is the most accurate. However, regressing the high-dimensional IBL directly may lead to local minimal and it is difficult to converge during training. With much fewer parameters, SH is a low-dimensional lighting representation and can be easily predicted with a compact decoder. Unfortunately, SH is too sensitive to its parameters’ variation. Moreover, the network has a tendency to overfit to the ringing artifacts when employing high-order SH functions to preserve high-frequency information.

SG is able to represent high-frequency lighting and avoid ringing artifacts. However, Gardner et al. [10] showed that it is hard to optimize the positions of SGs, especially when the number of SGs is large. In comparison, DSGLight represents an indoor panorama with a certain number of SG functions which have fixed positions and bandwidth. It is more stable and effective than predicting arbitrary and floating light sources. In addition, as the vertices are spread evenly on the unit sphere and cover the entire sphere, DSGLight can capture both dominated light sources and detailed environmental lighting. This allows high-quality AR rendering with consistent shading and shadows, as well as fine details on specular objects.

For a fair comparison, we visually compare the panoramas synthesized with 64 floating SGs and those synthesized with our DSGLight in Figure 6. As shown in (1), each floating SG is represented by three parameters: the centering direction, the bandwidth, and the RGB information. Therefore, 64 floating SGs have the same number of parameters with our DSGLight, which contains 128 SGs with the fixed directions and bandwidth. We retrieve 64 floating SGs following Gardner et al. [10], ensuring that SGs model every significant light source. We can see that most floating SGs are optimized to model the light sources on the ceiling since they provide the most light intensities. Consequently, 64 floating SGs are weak in representing detailed environment lighting. In addition, the unbalanced distribution of SGs is difficult for the network to regress. In contrast, DSGLight provides an elegant solution for the unbalanced distribution. It leverages more fixed SGs without additional parameters and achieves better representation ability.

### 3.5 Dataset preparation

To generate the dataset for training, we have to prepare a large number of images and their corresponding DSGLight parameters. Given an high dynamic range (HDR) panorama, generating ground truth DSGLight can be solved as an optimization problem, i.e., minimizing the difference between illumina-
Figure 7 (Color online) The overall pipeline of our method. Given an input image, we first leverage a backbone CNN as a feature extraction module (a) to learn the features of nodes in DSGLight. Then, we construct a graph with feature vectors initialized by the feature extraction module. A GCN (b) is employed to exchange feature vectors across adjacent nodes and finally predicts the values (color or depth) of the SG nodes.

The overall pipeline of our method:

\[
A^* = \arg\min_A \sum_{p \in P} \left( p - \sum_{i=1}^{N} G(F(p); a_i, \lambda_i, \mu_i) \right)^2,
\]

where \(A = \{a_1, a_2, \ldots, a_N\}\) is the set of lobe amplitudes of SGs, \(p\) is a pixel in the ground truth panoramas \(P\), and \(F\) maps \(p\) to a direction \(v\) in the sphere. When fixing \(P\), \(\mu\), and \(\lambda\), the problem boils down to a linear optimization problem, which can be solved easily by nonnegative least square algorithms.

We currently perform the optimization on two datasets:

- Laval dataset [11] contains 2200 high resolution indoor panoramas which are fully HDR without any saturation. This dataset also contains depth values for each panorama.
- SUN360 dataset [40] contains 12000 indoor low dynamic range (LDR) panoramas captured in a wide range of indoor scenes. For this dataset, we first convert each LDR panorama into an HDR panorama with the trained network of [41]. This dataset does not contain depth values. Therefore, we can only handle DSGLight with RGB channels when using examples from this dataset.

We further employ Mitsuba renderer [42] to sample several FoV-limited images from each panorama at random elevation between \(-20^\circ\) and \(20^\circ\), azimuth between \(-180^\circ\) and \(180^\circ\) and FoV between \(60^\circ\) and \(80^\circ\). Meanwhile, the corresponding panorama is captured by aligning its center with the view direction.

4 Deep graph learning for DSGLight

Our goal is to recover DSGLight of the environment in which the photograph has been taken. It does not simply assign light intensity and depth value to each visible light source, but must extrapolate large portions of the invisible illumination. Thus, it requires wide context information and must exploit very specific relationships between light sources to guide estimation. To this end, we exploit a GCN to use prior knowledge on the non-Euclidean nature of graph structure of DSGLight and cope with significant amounts of potential relationships between nodes. Running graph convolutions updates the indoor-specific perceptual features extracted from the input image, which is equivalent as driving light sources’ intensities towards the correct light intensities in the RGB color space.

4.1 Network architecture

Our framework consists of two parts: a feature extraction module and a GCN. Figure 7 shows the architecture of our network.

In the feature extraction module, we can use any CNN-based backbone to extract the features of an input image. Currently, we adopt the VGG-19 [43] as the backbone and generate a \(512 \times 7 \times 11\) feature matrix given an image with a resolution of \(240 \times 360\). This feature matrix is further connected to GCN with two fully-connected layers, which transform the feature map to the features of each node in DSGLight.

In our framework, a graph convolutional layer takes a graph as input and outputs a graph with a different feature matrix. It can be written as the following non-linear function:

\[
H^{l+1} = \sigma(\mathbf{E}H^l\mathbf{W}^l),
\]
where $H^l$ is the input of the $l$th layer, $W^l$ is a trainable weight matrix, $E$ is the adjacency matrix describing the graph structure in the matrix form, and $\sigma$ is a non-linear activation function (ReLU in our experiments). Stacking graph convolutional networks can combine graph node features and graph topological structural information to make final predictions.

GCN layer takes the node representations $H^l$ from previous layer as inputs and outputs new node representations. Thus we construct the graph $G = \{V, E\}$ as the input of GCN. GCNs are vertex-based and perform convolution on vertices with edges denoting connections. $V$ is the set of nodes corresponding to vertices in DSGLight. $F$ is the features of vertices learned from CNN. $E$ is the adjacency matrix in (5). To get the adjacency matrix, we use k-nearest neighbors (k-NN) to construct the directed dynamic edges between nodes.

Our GCN module comprises 4 graph convolutional layers. Li et al. [44] stated that stacking more layers into a GCN would easily lead to the vanishing gradient problem and over-smoothing. Our experiments show that four GCN layers are sufficient for multi-scale feature extraction and are stable for training.

4.2 Loss functions

In the design of loss functions, we handle color and depth separately. For depth, we only use L2 loss. For color, we design a hybrid loss that consists of a reconstruction loss $L_R$, a perceptual loss $L_{VGG}$ and a weighted L2 loss $L_W$ for the SG parameters, i.e.,

$$\mathcal{L} = L_W + \alpha L_R + \beta L_{VGG},$$

where $\alpha$ and $\beta$ are the weights to balance different losses. In our training, we set $\alpha$ and $\beta$ to 0.2 and 0.1, respectively.

Reconstruction loss penalizes the per-pixel intensity distance between the ground truth and the predicted environment map reconstructed with (2). The perceptual loss aims to enhance the perceptual similarity in the predicted illumination. We adopt a VGG-19 architecture which is pre-trained in the image classification task. Let $\Phi$ be the output of the last pooling layer in VGG-19, and $N_\Phi$ represent the total number of pixels in this feature space. The perceptual loss is defined as

$$L_{VGG}(p, \hat{p}) = \frac{1}{N_\Phi} \ell_2(\Phi(p), \Phi(\hat{p})), \quad (7)$$

where $p$ and $\hat{p}$ denote the ground truth and the predicted environment map, separately.

L2 loss is used mostly in regression problems but it will lose high-frequency information in training. Since the DSGLight has a wide range of values, we reweight SGs in the DSGLight to boost the importance of salient SGs. Inspired by ACESFilm (Academy Color Encoding System) curve, which distributes the weights to pixels of HDR images empirically and achieves good tone mapping result, we use the following weighted function:

$$f_w(x) = \frac{x(2.51x + 0.03)}{x(2.43x + 0.59) + 0.14} \quad (8)$$

The weighted L2 loss for SG parameters is calculated as

$$L_W(A, \hat{A}) = f_w(A)\ell_2(A, \hat{A}), \quad (9)$$

where $A$ and $\hat{A}$ represent the ground truth and our predicted parameters, respectively.

4.3 Discussion on the locality of indoor lighting and spatially-varying rendering

In general, lighting estimation models [11,14,15] reconstruct a full panorama from a single FoV-limited image. The panorama encodes the directional distribution of lighting around the current point of view. Unfortunately, most light sources in indoor scenes have a local influence as illustrated in Figure 8. Directional lighting (Figure 8(a)) makes the rendering of the virtual object irrespective of its spatial position. This contradicts the real scenario of indoor lighting in which the shading of the virtual object varies spatially with respect to their related positions, as shown in Figure 8(b). To address the problem, we augment DSGLight with depth values. Specifically, the lobe amplitude parameters of each SG contain four channels (three for color and one for depth) in DSGLight for an input image. We set the distance from the SG to the viewpoint as the predicted depth value. In this case, each SG can be regarded as
Figure 8 (Color online) Comparison of rendering between directional lights and local area lights. Our depth-augmented DSGLight model with local area lights enables spatially-varying rendering of inserted virtual objects and also avoids temporal flickering when the objects move. (a) Directional lights and rendering results; (b) SG-shaped area lights and rendering results.

an area light source with a full 3D coordinate. In contrast to directional lights in the environment map, area light sources allow us to render the object with spatially-coherent illumination at any location in a scene. As the object moves in the scene, the direction vector and distance between the object and each SG can be correctly calculated by their coordinates. Hence, the direction of the shadow generated by the object and the correct attenuation value of each SG can be changed according to the movement of the object, which effectively improves the realism of rendering.

Training DSGLight with depth values requires a large number of panoramas and their ground truth depth maps. However, among our two datasets, only Laval dataset has manually annotated per-pixel depth. From these annotated depth maps, we optimize the SGs of depth according to (4) where \( P \) is replaced by the depth map.

Some previous methods, such as Li et al. [45] and Garon et al. [12], also try to handle spatial variations of lighting for indoor scenes. Unlike ours, they estimate indoor lighting for each inserted point independently, yielding a local representation of lighting that varies spatially. Although they produce finer spatial light prediction, their strategy easily incurs temporal flickering when the inserted virtual object moves in the scene. In contrast, our global representation of indoor lighting guarantees temporal consistency for moving objects.

4.4 Training details

Our network is implemented on top of Tensorflow [46]. We train it using SGD and the Adam solver with the moment parameters \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \). The learning rate is initially set to 0.001 and halves every 40 epochs. The weights of all layers in our network are initialized with the Xavier uniform. Training examples are fed into our network with a mini-batch size of 5. We train the network for 150 epochs which takes about two days on one NVIDIA V100 GPU.

5 Evaluation

In this section, we evaluate the performance of our method in indoor illumination prediction and make comparisons with the state-of-the-arts.

5.1 Dataset and metrics

To generate our dataset, HDR panoramas are subdivided into two parts, one for training and the other for testing. Therefore, the panoramas in the test set will not appear during training. As mentioned in Subsection 3.5, we then randomly sample FoV-limited images and generate the ground truth DSGLights from the panoramas. The SUN360 dataset contains 50000 images for training and the test set contains 2000 images. The Laval dataset has 10000 images for training and 2000 for testing. Visual comparisons are conducted on both predicted environment maps and rendered images with virtual object insertion. PSNR is adopted as the quantitative measure.

5.2 Comparisons to previous methods

We predict indoor illumination with different methods and render two scenes in Figures 9 and 10, respectively. For Gardner et al. [10] and Zhan et al. [15], we implemented and trained their networks with two datasets separately. Lighthouse [14] requires a held-out perspective view near the input stereo pair and the depth. Since SUN360 lacks ground truth depth, we only retrain Lighthouse on the Laval dataset. As for evaluation on SUN360, we use their pre-trained model. Rendering is performed with a physically-based Mitsuba renderer [42]. The model of Gardner et al. [10] contains 11.7 M trainable parameters and
Lighthouse [14] has 13 M parameters. The state-of-the-art method, Zhan et al. [15], leverages a much larger network with 133.7 M trainable parameters. It costs 130 ms to infer the lighting for an image on an NVIDIA GeForce GTX 1080Ti. With 144.6 M trainable parameters, our DSGLight is comparable with Zhan et al. [15] and we can process an image in 136 ms.

**Qualitative evaluation on the Laval dataset.** In Figure 9, we render the Armadillo scene using the predicted illumination of four recent methods, as well as the ground truth and our method. The

**Figure 9** (Color online) Rendering the Armadillo scene (purely diffuse) with the indoor illumination predicted by different methods. Quantitative metrics in terms of PSNR for the rendered images are displayed below each method.
Figure 10 (Color online) Rendering the Dragon scene (specular) with the indoor illumination predicted by different methods. Quantitative metrics in terms of PSNR for the rendered images are displayed below each method. Depth values are not concerned in these cases.

renderings of the Armadillo on a plane highlight the accuracy of estimated light directions. Note that our generated shadows and ambient lighting are closer to the ground truth, as compared with other methods.

Gardner et al. [11] proposed a CNN-based pipeline to predict an RGB panorama and a light mask for the indoor illumination. The predicted panorama is usually very smooth and may have color drift. Therefore, the synthesized images are quite different from those generated by the ground-truth lighting.
Table 2  Quantitative comparisons in terms of PSNR (dB). The best scores are highlighted in bold. -GCN and -LW represent the network trained without GCN module and without weighted L2 loss, respectively.

| Dataset | Method        | Map     | Armadillo | Dragon |
|---------|---------------|---------|-----------|--------|
| Laval   | Gardner et al. [11] | 14.45   | 20.60     | 18.35  |
|         | Gardner et al. [10] | 16.49   | 22.21     | 21.14  |
|         | Lighthouse [14]   | 17.50   | 21.42     | 23.00  |
|         | Zhan et al. [15]  | 13.80   | 16.63     | 18.76  |
|         | Ours            | 22.54   | 26.16     | 24.00  |
|         | -GCN           | 20.90   | 24.83     | 22.11  |
|         | -LW            | 21.87   | 20.30     | 22.61  |
| SUN360  | Gardner et al. [11] | 11.90   | 23.83     | 20.15  |
|         | Gardner et al. [10] | 12.36   | 23.58     | 19.04  |
|         | Lighthouse [14]   | 13.63   | 21.10     | 21.04  |
|         | Zhan et al. [15]  | 9.02    | 20.17     | 15.97  |
|         | Ours            | 13.75   | 24.82     | 21.81  |
|         | -GCN           | 13.16   | 23.39     | 21.21  |
|         | -LW            | 13.70   | 16.85     | 21.54  |

Gardner et al. [10] used a parametric form based on SGs to encode indoor light sources, but the centers of the light sources are allowed to vary wildly, leading to unstable results. For the method of Srinivasan et al. [14], it fails to predict the lighting distribution outside the known regions, leading to a large bias on rendering. Zhan et al. [15] have proved the effectiveness of SGs by generating accurate illumination maps under the guidance of predicted SG parameters. However, they do not support spatially-varying lighting prediction. Most importantly, their CNN networks do not explore the long-range correlation between SGs with a limited receptive field.

Qualitative evaluation on the SUN360 dataset. In Figure 10, we further compare these methods by rendering a specular Dragon model lit by the estimated lighting. Since the model is highly specular, it can clearly reflect the environment around it. As expected, our method captures details in the surroundings (e.g., the white wall and the red floor) quite well. The predicted panorama is plausible and the details provide better visual coherence. In comparison, Gardner et al. [11] captured the hue of the scene, but the lighting distribution is overly smooth. In contrast, Zhan et al. [15] hallucinated many details different with the ground truth. For the method proposed by Gardner et al. [10], it hardly infers and represents lighting details.

Quantitative evaluation. We also provide the quantitative metrics in terms of PSNR for each method in Table 2. In this table, we list the average PSNR values for the estimated environment map, the rendered Armadillo scene and the Dragon scene. Obviously, our method achieves higher PSNR values than others under every testing scenario. This further shows that our method achieves state-of-the-art performance on indoor lighting prediction.

5.3 Ablation study

To validate the importance of graph convolutional layers, we replace the GCN module with a fully connected layer in the whole pipeline. We define this variant model as -GCN. For a fair comparison, -GCN model has 139.2 M trainable parameters, which is close to that of DSGLight.

Figure 11 reports the performance of these models. We observe that environment maps generated by the model without GCN module lack fine details for the scenes. It cannot exchange information across neighboring vertices through GCN layers and vertices work independently. Therefore, it is hard to infer illumination from the known region of panoramas. Table 2 shows that PSNR drops slightly when our network replaces the GCN module. It also shows that weighted L2 loss is beneficial for improving the performance.

5.4 Validation of spatially-varying lighting

As aforementioned, our method supports spatially-varying lighting when inserting virtual objects into different locations of an indoor scene, due to the availability of the depth information. To demonstrate this, we insert virtual objects at different locations in real images of [12]. Two examples are provided in Figure 12. Here, we compare our method with [45] that also enables spatially-varying indoor lighting.
As seen, our results are closer to the ground truth. More importantly, our method guarantees temporal consistency of lighting when the virtual object moves in the scene, since our DSGLight is unique and global for a given image. However, Li et al.’s method [45] estimates local lighting independently for each inserted position, which may easily cause temporal flickering.

To further show the superiority of our method, we conduct a perceptual user study on 20 scenes of [12]. For each scene, we generate a reference composite by relighting a bunny model with its ground truth light probe. We compare these results with objects that were relit with estimated illumination from [45] and our method. Users are asked to pick which rendering is more realistic between image pairs rendered with ground truth and the estimated lighting. We gather responses from 138 unique participants and the results are shown in Figure 13. Both [45] and our method beat each other in half of the scenarios, which suggests that they are of equal visual quality. However, our method outperforms [45] slightly (32.00% vs. 30.75%) on the average confusion.

5.5 More results on virtual object insertion

In Figure 14, we show more examples of inserting virtual objects in one scene. Given an input image, we predict the lobe amplitude parameters of each SG which contains four channels (three for color and one for depth) in DSGLight. During rendering, each SG acts as an SG-shape area light source and its 3D position and light intensity are determined by SG parameters. Clearly, adding depth to local illumination can automatically adjust the direction of shadows and the highlights on the surface of inserted virtual objects. This significantly improves the realism of rendering.
Figure 13 (Color online) Each scene in the user study is shown as a column, where different colors indicate that users preferred the corresponding method instead of the ground truth.

Figure 14 (Color online) More results on virtual object insertion. In each group, the left shows the input and the right shows the output after virtual object insertion.

Figure 15 (Color online) Predictions of DSGLight (Columns 3, 5 and 7) for input images (Columns 2, 4 and 6) which are randomly captured from the same scene (Column 1).

5.6 Limitations

Despite the state-of-the-art performance, our method still has some limitations. First, DSGLight is an approximation to the detailed environment map, lacking sharp features. This prevents specular objects from generating sharp textures reflecting the surrounding. Second, DSGLight is robust for the input images with overlap between them. However, the predictions could vary considerably between independent images since DSGLight processes input images individually. In Figure 15, we randomly sample three images in the same scene and show the predictions of DSGLight. We can see that DSGLight is robust to infer light sources in the sampled scene. When the input image (Column 6 in Figure 15) is totally different, DSGLight outputs a reasonable environment map that is only consistent with the input. Third, the out-of-view depth generated by our method is not very accurate due to the high ill-posedness. This would be solved by multiple input images.
6 Conclusion

In this paper, we have proposed the first graph learning-based strategy for inferring high-quality and spatially-varying indoor illumination from a single image. To represent both dominated local lights and detailed environmental lighting with high fidelity, we encode the panoramic indoor illumination in DSGLight, a depth-augmented lighting model with 128 SGs evenly distributed over a sphere. A GCN is adopted to predict non-Euclidean DSGLight from a 2D image. Through thorough comparisons, we have demonstrated that our method outperforms previous state-of-the-art methods both qualitatively and quantitatively.

Acknowledgements  This work was supported by National Natural Science Foundation of China (Grant Nos. 62032011, 61972194).

References

1. Einabadi F, Guillemaut J, Hilton A. Deep neural models for illumination estimation and relighting: a survey. Comput Graphics Forum, 2021, 40: 315–331
2. Liu Y, Gevers T, Li X Q. Estimation of sunlight direction using 3D object models. IEEE Trans Image Process, 2015, 24: 952–942
3. Hold-Geoffroy Y, Sunkavalli K, Hadap S, et al. Deep outdoor illumination estimation. In: Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 2373–2382
4. Zhang J, Sunkavalli K, Hold-Geoffroy Y, et al. All-weather deep outdoor lighting estimation. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 10150–10158
5. Hold-Geoffroy Y, Athawale A, Lalonde J. Deep sky modeling for single image outdoor lighting estimation. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 6920–6928
6. Yu P, Guo J, Huang F, et al. Hierarchical disentangled representation learning for outdoor illumination estimation and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 15313–15322
7. Barron J, Malik J. Intrinsic scene properties from a single RGB-D image. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013. 38
8. Maier R, Kim K, Cremers D, et al. Intrinsic3D: high-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In: Proceedings of IEEE International Conference on Computer Vision, Venice, 2017. 3133–3141
9. Wei H, Prévost D, Lalonde J. Learning to estimate indoor lighting from 3D objects. In: Proceedings of International Conference on 3D Vision, Verona, 2018. 199–207
10. Gardner M, Hold-Geoffroy Y, Sunkavalli K, et al. Deep parametric indoor lighting estimation. In: Proceedings of 2019 IEEE/CVF International Conference on Computer Vision, Seoul, 2019. 7174–7182
11. Gardner M A, Sunkavalli K, Yumer E, et al. Learning to predict indoor illumination from a single image. ACM Trans Graph, 2017, 36: 1–14
12. Garon M, Sunkavalli K, Hadap S, et al. Fast spatially-varying indoor lighting estimation. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 6901–6910
13. Song S, Funkhouser T A. Neural illumination: lighting prediction for indoor environments. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 6918–6926
14. Srinivasan P P, Mildenhall B, Tancik M, et al. Lighthouse: predicting lighting volumes for spatially-coherent illumination. In: Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, 2020. 8077–8086
15. Zhan F, Zhang C, Hu W, et al. Sparse needles for lighting estimation with spatial transport loss. In: Proceedings of the IEEE International Conference on Computer Vision, 2021
16. Cheng D, Shi J, Chen Y, et al. Learning scene illumination by pairwise photos from rear and front mobile cameras. Comput Graphics Forum, 2018, 37: 213–221
17. Green R. Spherical harmonic lighting: The gritty details. 2003. https://api.semanticscholar.org/CorpusID:116856600
18. Liu B, Xu K, Martin R R. Static scene illumination estimation from videos with applications. J Comput Sci Technol, 2017, 32: 430–442
19. Tsai Y T, Shih Z C. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. ACM Trans Graph, 2006, 25: 967–976
20. Wang J, Ren P, Gong M, et al. All-frequency rendering of dynamic, spatially-varying reflectance. ACM Trans Graph, 2009, 28: 133
21. Xu K, Sun W L, Dong Z, et al. Anisotropic spherical Gaussians. ACM Trans Graph, 2013, 32: 1–11
22. Wu C, Wilburn B, Matsushita Y, et al. High-quality shape from multi-view stereo and shading under general illumination. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011. 969–976
23. Kurach K, Hesla I, Forsyth D, et al. Rendering synthetic objects into legacy photographs. ACM Trans Graph, 2011, 30: 1–12
24. Khan E A, Reinhard E, Fleming R W, et al. Image-based material editing. ACM Trans Graph, 2006, 25: 654–663
25. Wei X, Chen G, Dong Y, et al. Object-based illumination estimation with rendering-aware neural networks. In: Proceedings of the 16th European Conference on Computer Vision, Glasgow, 2020. 380–396
26. Arwood J, Tosley D. Diffusion-convolutional neural networks. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016. 2014–2023
27. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, Red Hook, 2016. 3844–3852
28. Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017. 1025–1035
Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations, Toulon, 2017.

Shuman D I, Narang S K, Frossard P, et al. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process Mag. 2013, 30: 83–98.

Bastings J, Titov I, Aziz W, et al. Graph convolutional encoders for syntax-aware neural machine translation. 2017. ArXiv:1704.04675.

Marcheggiani D, Titov I. Encoding sentences with graph convolutional networks for semantic role labeling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence, 2018. 7444–7452.

Li Y, Ouyang W, Zhou B, et al. Factorizable net: an efficient subgraph-based framework for scene graph generation. In: Proceedings of European Conference on Computer Vision, 2018. 349–363.

Qi X, Liao R, Jia J, et al. 3D graph neural networks for RGBD semantic segmentation. In: Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV), 2017. 5209–5218.

Yang J, Lu J, Lee S, et al. Graph R-CNN for scene graph generation. 2018. ArXiv:1808.00191.

Xiao J, Ehinger K A, Oliva A, et al. Recognizing scene viewpoint using panoramic place representation. In: Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, 2012. 2695–2702.

Eikertsen G, Kronander J, Denes G, et al. HDR image reconstruction from a single exposure using deep CNNs. ACM Trans Graph, 2017, 36: 1–15.

Nimier-David M, Vicini D, Zeltner T, et al. Mitsuba 2: a reconfigurable forward and inverse renderer. ACM Tran Graph, 2019, 38: 1–17.

Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the 3rd International Conference on Learning Representations, San Diego, 2015.

Li G, Müller M, Thabet A K, et al. DeepGCNs: can GCNs go as deep as CNNs? In: Proceedings of 2019 IEEE/CVF International Conference on Computer Vision, Seoul, 2019. 9266–9275.

Li Z, Shafiei M, Ramamoorthi R, et al. Inverse rendering for complex indoor scenes: shape, spatially-varying lighting and SVBRDF from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 2475–2484.

Abadi M, Agarwal A, Barham P, et al. TensorFlow: large-scale machine learning on heterogeneous systems, 2015. arXiv:1603.04467.