Spatially-Varying Outdoor Lighting Estimation from Intrinsics

Yongjie Zhu† Yinda Zhang Si Li* Boxin Shi3, 4 *
1School of Artificial Intelligence, Beijing University of Posts and Telecommunications
2Google 3NELVT, Department of Computer Science and Technology, Peking University
4Institute for Artificial Intelligence, Peking University

Abstract

We present SOLID-Net, a neural network for spatially-varying outdoor lighting estimation from a single outdoor image for any 2D pixel location. Previous work has used a unified sky environment map to represent outdoor lighting. Instead, we generate spatially-varying local lighting environment maps by combining global sky environment map with warped image information according to geometric information estimated from intrinsics. As no outdoor dataset with image and local lighting ground truth is readily available, we introduce the SOLID-Img dataset with physically-based rendered images and their corresponding intrinsic and lighting information. We train a deep neural network to regress intrinsic cues with physically-based constraints and use them to conduct global and local lightings estimation. Experiments on both synthetic and real datasets show that SOLID-Net significantly outperforms previous methods.

1. Introduction

Estimating outdoor lighting from a single image is one of the fundamental problems in computer vision. By providing outdoor scene properties from the physical aspect, it has huge impact on many applications, e.g., face/body relighting, scene understanding, augmented reality (AR), and so on. This task is rather challenging since images are formed by conflating lighting with complex surface reflectance distribution and object geometry. In the outdoor scenario, existing solutions usually employ low-dimensional parametric models such as the Hošek-Wilkie (HW) sky model [11] with four parameters to fit the sky illumination. The capacity of parametric models is not sufficient to represent the complex real-world illumination, and a recent non-parametric approach using an autoencoder to learn the sky illumination model from a large-scale sky panorama dataset and encoding the lighting information from a single limited Field-of-View (FOV) image shows more promising results [10].

However, as far as we know, all existing outdoor lighting estimation methods [11, 10, 26] only consider the outdoor illumination as a single global map without any spatially-varying consideration, i.e., the light probe is surrounded by an environment map that casts rays from infinitely far away. A spatially-varying lighting estimation has proved to be successful in indoor scenarios, which is achieved by modeling local indoor lighting using low-frequency parametric lighting represented by spherical harmonics (SH) [9, 5] or panoramic environment map [20].

Extending spatially-varying lighting estimation from indoor to outdoor is non-trivial in three aspects: 1) The extremely high-dynamic-range (HDR) sunlight and the complicated sky light under different weather conditions make it more difficult to parameterize outdoor than indoor lighting [9, 5], while the existing non-parametric sky model [10] treats it as a pure deep learning task without considering physics image formation constraint. 2) Non-parametric
spatially-varying local lighting estimation is highly ill-
posed, since different 3D locations should have different
lighting observations and the majority of local observation
is missing [20]. 3) HDR and panoramic images capturing
local lighting and geometry information in outdoor are not
yet available, despite there are many datasets for such a
purpose in the indoor scenario by synthetically generating
scenestion from SUNCG [21] and Matterport3D [4].

In this paper, we propose SOLID-Net, a neural net-
work for Spatially-varying Outdoor Lighting estimation us-
ing cues from Intrinsic image Decomposition, as shown
in Figure 1. We tackle the three major challenges mentioned
above by proposing a two-stage framework: 1) We train a
single-in-multi-output CNN to decompose an input image
into intrinsic parts: albedo (material-related), normal and
plane distance (geometry-related), and shadow (lighting-
related). These intrinsics provide a physically-based shad-
ing constraint by fitting SH-represented global lighting with
low-frequency information, which is then combined with
extracted sky features from the input image to generate a
non-parametric sky model like [10]. 2) With the estimated
gometry from decomposed intrinsics, we further warp the
input image and estimated shadow map with limited FOV to
a spherical projection centered at the target location, which
provides panoramic observation to reduce the ill-posed is-
ue. This is then combined with global sky lighting from the
previous step as input to train a multi-input-single-output
CNN for complementing high-frequency local lighting esti-
imation. 3) We use the Blender SceneCity [2] to create city
models that contain a large set of outdoor scenes and render
a synthetic outdoor lighting estimation dataset with labeled
location information and corresponding lighting effects us-
ing a physically-based path-tracer to facilitate the training
of our network. SOLID-Net demonstrates significant im-
provements over other methods by making contributions in

- integrating shading constraint from intrinsic decompo-
sition into the global sky lighting estimation;
- producing high-frequency local lighting estimation via
panoramic warping and shadow map reference; and
- building the first spatially-varying outdoor lighting es-
timation dataset with ground truth labels.

2. Related Work

Outdoor lighting estimation. Stumpfle et al. [23] pro-
posed to explicitly capture the HDR outdoor lighting envi-
ronments that include the sun and sky with multiple expo-
sures. Lalonde et al. [12] first proposed lighting estimation
from a single, generic outdoor scene. Their approach relied
on multiple cues (such as shadows, shading, and sky ap-
pearance variation) extracted from the image. There are sol-
lutions using parametric models to represent outdoor light-
ing: Cheng et al. [6] estimated lighting from the front and
back camera of a mobile phone. However, they represented
lighting using low-frequency SH, which does not appro-
priately model outdoor lighting. Hold-Geoffroy et al. [11]
learned to estimate Hošek-Wilkie (HW) sky model param-
eters from a single image, which is further extended by
Zhang et al. [26] with a more flexible parametric Lalonde-
Matthews (LM) sky model. To include more information
of the sky, Hold-Geoffroy et al. [10] designed an autoen-
coder to learn a non-parametric sky model from a large sky
panorama dataset [14] and trained a network to learn the sky
lighting from limited-FOV images. LeGendre et al. [15]
used a mobile phone camera with three different reflective
spheres to capture lighting ground truth and used these data
to train their deep model effectively, but these spheres are
still global lighting probes.

Local lighting estimation. A direct way of obtaining the
local lighting of an environment is to capture the lighting
intensity at a target location using a probe of known shape.
Debevec et al. [7] showed that HDR environment maps can
be captured with a reflective metallic sphere captured with
the scene. Barron an Malik [5] decomposed the scene into
intrinsinc components including spatially-varying SH-based
lighting, but it required an RGBD image as input and re-
lied on hand-crafted priors. To learn local lighting represen-
tation, Garon et al. [9] predicted fifth-order SH coeffi-
cients from an input image and local patches with synthetic
data. In more recent progress, Li et al. [16] proposed a
dense spherical Gaussian lighting representation with dif-
ferrntiable rendering to conduct scene editing. But all the
methods mentioned above only considered indoor paramet-
ric lighting and are hard to be extended to outdoor light-
ing. Song et al. [20] proposed a cascaded model (denoted
as NeurIllum for brevity) to recover high-frequency local
lighting with warped color image according to recovered
geometry, which showed promising texture details but the
lighting positions are sometimes less accurate due to the
loss of massive information in the panorama.

3. Dataset

A large dataset containing HDR images and their cor-
responding illumination measured at different locations in
a scene is required to learn to estimate outdoor intrinsics
and local lightings. Existing outdoor panorama datasets,
such as [10, 27] only provide a single global illumination
map assuming distant lighting, which cannot be used to
learn local lighting estimation. To provide training data for
solving “SOLID” problem, we introduce the SOLID-Img,
a dataset for Spatially-varying Outdoor Lighting estima-
tion with ground truth Intrinsic Decomposition labels and
a large amount of rendered Images as shown in Figure 2.
3.1. Data Generation

We adopt 3D city models from the Blender SceneCity [2] to create synthetic scenes. In Blender SceneCity, there are 450 unique objects in 80 material categories. The object models provide surface materials, including diffuse albedo, roughness, and transparency, which are used to obtain photo-realistic renderings.

Camera setting. For each road block, we select a set of cameras with diverse views seeing most objects in the context, to provide comprehensive information for lighting estimation, as shown in Figure 2(a). Our process starts by selecting the “best” camera [28] for each of the six horizontal view direction sectors in every road block. For each sector, we select the view with the highest percentage pixel coverage according to item buffer, as long as it has more than three object categories.

Scene rendering. We collect 70 HDR environment maps from HDRI Haven [3] which cover different solar zenith angles from sunrise to sunset. To simulate different sunlight directions, we rotate each HDR environment map along the latitude direction with a random angle sampled uniformly in $[0^\circ, 60^\circ]$. Then we render images using the camera settings above and these HDR environment maps with the physically-based Blender Cycles rendering engine [1], to generate photo-realistic renderings. The resolution is set as $320 \times 240$ with a physically-based path tracer of 512 samples. We record the material buffer (diffuse albedo buffer, normal buffer, depth buffer) as intermediate ground truth. We represent 3D geometry using the surface normal and plane distance, and render both as suggested in [22]. To render shadows, we set the whole scene as a single Lambertian material and render it twice with shadow turned on and off respectively, from which shadow maps are calculated by checking the difference, as shown in Figure 2(b).

Local lighting collections. To obtain the ground truth of global lighting, we save the rotated environment maps with $256 \times 128$ solution. To collect local lighting, we randomly sample 4 locations in the scene to render 4 local light probes. The image is split into 4 quadrants, and a random 2D coordinate is sampled uniformly in each quadrant (excluding the sky part and 5% pixels near the image boundary). The 3D centers of local cameras are calculated by casting a ray from the camera recording the scene to the surface of the scene and getting the first intersection point. From that point, we move the local camera center 10cm away along the plane surface normal to prevent large invalid pixels and render a local light probe at this position with $256 \times 128$ resolution. All local light probes are rendered in the equirectangular representation, as shown in Figure 2(c).

3.2. Data Filtering

Inspired by [28], we remove low-quality renderings that are with different color distribution with natural images, e.g., with overly low or high intensities. To obtain a prior color distribution on real images, we compute normalized color histogram for 1100 selected real images from the Google Street View Dataset [25]. For each rendered image, we calculate the histogram similarity with those from Google Street View as the sum of minimal value of each bin; and then we assign it with a score calculated as the largest histogram similarity by comparing it to all real images; finally, we select all images with color similarity score larger than 0.7, as shown in Figure 2(d). This process selects 38000 images from the initially rendered image set, composing the SOLID-Img dataset. Then care is taken to split the dataset into the train/test set according to different lighting conditions.

4. Method

This section introduces the design methodology of SOLID-Net whose pipeline is shown in Figure 3. It is a two-stage framework that learns to reconstruct locally HDR outdoor environment maps, trained with the SOLID-Img dataset introduced in Section 3.

4.1. Problem Formulation

We formulate illumination estimation as a regression problem. Given an LDR image $I$ with limited FOV and a selected pixel location “$i$” in homogeneous coordinate $(x_0, y_0, 1)^T$, our model outputs an HDR illumination $H_i$ centered around the 3D location of the pixel “$i$” and a global
4.2. Network Architecture

A straightforward approach to estimating the outdoor lighting from the scene would be to simply take the single limited FOV image as input, encode it into a feature map using a CNN, and feed the feature map into a lighting-regression sub-network [8, 10]. Unsurprisingly, we find it results in outdoor lighting estimation with higher error (see Figure 6), presumably because it is difficult for the network to understand how to extract full FOV lighting from a limited FOV image. One way to improve it is to bring regularization from the Lambertian rendering equation [24], which however is challenging for outdoor spatially-varying lighting estimation because: 1) Outdoor scenes have large areas of shadow occlusion which cannot be directly fitted by the Lambertian model. 2) SH lighting has a limited dynamic range and it is too smooth to represent sharp sky lighting and detailed texture. Therefore, we propose a two-stage framework to jointly solve these problems by: 1) proposing an intrinsic image decomposition network (denoted as I-Net) that takes a limited FOV image as input and estimates its intrinsic components as well as a global sky environment map and 2) designing a panoramic completion module (denoted as P-Net) that estimates local lighting from outputs in the previous stage and the input location.

I-Net. As shown by blue blocks in Fig 3, I-Net takes a single limited-FOV LDR image \( I \) as input and various outputs including diffuse albedo \( A \), surface normal \( N \), plane distant map \( G \) [22], shadow map \( S \), and second-order SH coefficients \( P_{SH} \), and the sky environment map \( H_g \) generated from SH coefficients and sky features. We use a single encoder to capture global features of intrinsic information, and then use five decoders for \( A, N, G, S, \) and \( P_{SH} \) followed by a decoder lighting branch for sky environment map regression. Skip links are used for preserving details. In particular the sky map regression, we use a fully-connected (FC) layer to process the output feature maps of lighting branch encoder to generate a latent vector of size 27 (second-order SH in RGB). For the decoder, we reshape this vector and upsample it 8 times, and then combine it by flipping padded sky features to generate a 256\( \times \)128 HDR sky environment map. The lighting information encoded in \( P_{SH} \) can be considered as the low-frequency form of \( H_g \), and it is used to guide the recovery of the high-frequency sky environment maps with sky features extracted from the input images. In summary, I-Net predicts intrinsic components (examples are provided in Figure 2) and sky environment map:

\[
\{ \tilde{A}, \tilde{N}, \tilde{G}, \tilde{S}, \tilde{H}_g(\tilde{P}_{SH}) \} = I\text{-Net}(I). \tag{1}
\]

P-Net. The 3D location is calculated by the predicted normal vector \( n \) in \( N \) and plane distance \( p \) in \( G \) by I-Net for each pixel. If the camera intrinsic matrix is fixed as \( K = \begin{bmatrix} f_x & c_x; 0, f_y, c_y; 0, 0, 1 \end{bmatrix} \) of the whole image are provided, we can reproject them as a 3D scene \( \mathbf{P} = (x, y, 1)^T \) by \( \mathbf{P} = -\mathbf{v}^T \mathbf{n}, \) where \( \mathbf{v} = \begin{bmatrix} x - c_x / f_x; y - c_y / f_y; 1 \end{bmatrix}^T. \)

By using \( \mathbf{P} \) we warp the input image \( I \) and estimated shadow map \( S \) and spatially align them with the output local lighting to provide panoramic observation. First, we
compute the camera location according to the input point position \( I \) and apply 10cm translation (defined in Section 3) along the normal direction of the supporting plane to align with training data. Second, we perform a panoramic warping through a forward projection using the estimated geometry and camera location to map pixels in \( I \) and \( S \) as panoramic images (an example is provided in Figure 5). The Z-buffer is computed to discard invisible points and the points without projected positions are set to 0.

Since the local lightings share the same camera rotation, the sky parts in local lighting should be consistent, this motivates us to take the sky as an input to P-Net. As shown by orange blocks in Fig 3, P-Net concatenates the two incomplete panoramic image \( \tilde{I}_{warp} \) and \( \tilde{S}_{warp} \) and the global lighting estimated by I-Net \( \tilde{H}_g \) as inputs and outputs a dense pixel-wise prediction of local lighting panorama with full FOV and high-frequency details, as

\[
\tilde{H}_1 = \text{P-Net}(\tilde{I}_{warp}(1), \tilde{S}_{warp}(1), \tilde{H}_g).
\]

P-Net is implemented as a fully convolutional U-Net [18].

### 4.3. Loss Functions

**Direct supervision loss.** Direct supervision \( L_{sup1} \) for I-Net is provided to 1) diffuse albedo predictions via \( L_2 \) loss; 2) shadow predictions via \( L_2 \) loss; 3) surface normal predictions via cosine loss; 4) plane distance map predictions via \( L_1 \) loss, and 5) sky environment map predictions via \( L_1 \) loss. Then direct supervision \( L_{sup2} \) for P-Net is provided to local lighting predictions via \( L_1 \) loss.

\[
L_{sup1} = \|\mathbf{A} - \mathbf{A}\|_2 + \|1 - \mathbf{N} \cdot \mathbf{N}\|_2 + \|\mathbf{G} - \mathbf{G}\|_1
+ \|\mathbf{S} - \mathbf{S}\|_2 + \|\tilde{\mathbf{H}}_g - \mathbf{H}_g\|_1,
\]

\[
L_{sup2} = \|\tilde{\mathbf{H}}_1 - \mathbf{H}_1\|_1.
\]

where the \( \tilde{\cdot} \) means the estimations of I-Net and \( \cdot \) is the dot product for each vector in a matrix.

**Diffuse convolution loss.** In order to guide the sky environment map estimated by I-Net to extract low-frequency lighting information from the encoded SH coefficients, we add a diffuse convolution loss \( L_{dif} \) to force \( \mathbf{H}_g \) applied with the diffuse convolution to have a close appearance with a relighted pure Lambertian surface from \( \mathbf{P}_{SH} \):

\[
L_{dif} = \frac{1}{N} \sum_{i=1}^{N} \|D(H_g, i) - \text{diag}(\alpha_o) \cdot L_{b}([N_{\Omega}(i)]^{T})\|^2,
\]

where \( \alpha_o = [1, 1, 1]^{T} \) is the global diffuse albedo, \( L \in \mathbb{R}^{3 \times 9} \) is the SH coefficients by reshaping \( \mathbf{P}_{SH} \in \mathbb{R}^{1 \times 27} \), \( N_{\Omega} \) is the normal map of a sphere in panorama coordination and the second order SH basis is given by: \( b(n) = [1, n_x, n_y, n_z, 3n_x^2 - 1, n_x n_y, n_x n_z, n_y n_z, n_x^2 - n_y^2]^T \). \( D \) is the diffuse convolution function defined as

\[
D(H, i) = \frac{1}{K_i} \sum_{\omega \in \Omega_i} H(\omega) s(\omega)(\omega \cdot n),
\]

where \( \Omega_i \) is the hemisphere centered at pixel \( i \) on the global lighting environment map, \( n \) is the normal vector at pixel \( i \), and \( K_i \) is the sum of solid angles on \( \Omega_i \). \( \omega \) is a unit vector of direction and \( s(\omega) \) is the solid angle for a pixel in the panorama map of direction \( \omega \) with different scale factors (because pixels in the panorama map at different latitudes correspond to projections on the unit sphere with different area sizes).

**Inverse rendering reconstruction loss.** To make the network learn constraints from physically-based image formation model, we put SH coefficients as an intermediate variable and provide indirect supervision to \( \mathbf{P}_{SH} \) via an inverse rendering reconstruction loss \( L_{rec} \) on the directly illuminated part, by multiplying a non-shadowed mask to disregard the effect of shadows:

\[
L_{rec} = \|M_{\text{nes}} \odot (I_{\text{im}} - (\mathbf{A} \odot \mathbf{L}_{B}(\mathbf{N}))^{1/\gamma})\|_2,
\]

where \( \odot \) represents the element-wise product. We use a fixed \( \gamma = 2.2 \) to compress the dynamic range. \( M_{\text{nes}} \) is the non-shadowed mask computed using shadow maps from intrinsics; a binary Otsu segmentation on histogram of shadow maps is further used to eliminate weak interreflections; \( I_{\text{im}} \in \mathbb{R}^{3 \times K} \) is the RGB image matrix; \( \mathbf{A} \in \mathbb{R}^{3 \times K} \).
is the estimated diffuse albedo matrix; \( \mathbf{B}(\mathbf{N}) \in \mathbb{R}^{9 \times K} \) is a matrix stacked by \( \mathbf{b}(\mathbf{n}) \) which applied SH basis on the normal map.

**Tonemapped SSIM loss.** A structural similarity index measure (SSIM) loss between dynamic range compressed images with a fixed gamma parameter (2.2 in our experiment) is used to recover structure similarity between the estimation and ground truth:

\[
L_{\text{ton}} = \| \mathcal{S}(\mathcal{H} - (2^e \cdot \mathbf{H})) \|_{\text{SSIM}},
\]

\[
\mathcal{S}(\mathbf{H}) = \begin{cases} 
1 & \mathbf{H} > 1 \\
0 & \mathbf{H} < 0 \\
\mathbf{H} & 0 \leq \mathbf{H} \leq 1 
\end{cases}
\]

where \( e \) is the exposure intensity fixed as \(-0.3 \) in our experiments.

I-Net is trained by summing up direct supervision loss, diffuse convolution loss, and inverse rendering reconstruction loss as: \( L_{\text{I}} = L_{\text{sup1}} + L_{\text{diff}} + L_{\text{rec}} \), and then P-Net is trained by summing up direct supervision loss and tonemapped SSIM loss as: \( L_{\text{P}} = L_{\text{sup2}} + L_{\text{ton}} \).

5. Experiments

We perform detailed network analysis and present qualitative and quantitative results on our SOLID-Img test set. We also capture a small set of real LDR outdoor local environment maps to analyse the generalization of our method. Finally, we show relighted bunny results to validate our methods qualitatively\(^2\). To measure the accuracy of our predicted global sky environment map \( \mathbf{H}_g \) and local illumination maps \( \mathbf{H}_l \), we use mean absolute error (MAE) on the HDR sky environment map, angular error on the sun position and sun azimuth/elevation angles, and SSIM on the detailed local lighting as error metrics.

### 5.1. Analysis using Synthetic Dataset

**Effectiveness of I-Net.** To validate the design of intrinsic decomposition, we compare our global lighting estimation branch with three baseline models for the accuracy of estimated sun positions: 1) \( \text{LENet}_{\text{reg}} \) is a regression-based model that directly regresses the global sky from the input image. 2) \( \text{LENet}_{\text{ae}} \) is a two-stream convolution network used to regress sun azimuth angle and normalized HDR panorama from an LDR panorama [27]; we modify the input as a single limited-FOV image to adapt our task. 3) \( \text{LENet}_{\text{sky}} \) learns to estimate both the sun azimuth angle and a non-parametric sky [10]. In particular, \( \text{LENet}_{\text{ae}} \) learns azimuth estimation as a regression task, while \( \text{LENet}_{\text{sky}} \) treats it as a classification problem. All baseline models are retrained using SOLID-Img training dataset with the same setting\(^3\). Since our global lighting is represented by a non-parametric sky environment map, we compute the sun position by finding the largest connected component of the sky above a threshold (98%) and computing its centroid. And then we rotate estimated sky environment maps around their azimuth angles to make sure the sun is in the center of the image so that we can compare it with their baseline models.

From Figure 6, we can see that our method shows significant improvement than \( \text{LENet}_{\text{reg}} \) and \( \text{LENet}_{\text{ae}} \) and comparable improvement than \( \text{LENet}_{\text{sky}} \), thanks to the intrinsic cues. Qualitative results on the test dataset are shown in Figure 7\(^4\). Our relighting results and estimated lightings show a closer appearance to the ground truth (shown as insets) than other methods.

To help understanding how SH coefficients decode the global lighting information, we perform the Grad-CAM [19] on our global lighting encoder. We use the maximum response value of SH coefficients as the target backward label to find which regions of input are important for global lighting prediction. From Figure 8, the feature heatmaps validate that I-Net mostly captures directly illuminated information to estimate global lighting.

**Effectiveness of P-Net.** We train our P-Net with combinations of different inputs: warped incomplete LDR image

\(^2\)More results are in the supplementary material.

\(^3\)Detailed model structures are in the supplementary material.

\(^4\)Numerical results and MAE errors on estimated sky environment map are provided in the supplementary material.
As can be observed, our method can render correct lighting quality of synthetic object insertion to better illustrate the usefulness of spatially-varying outdoor lighting estimation. We conjecture it is because shadows provide occlusion information which is helpful for lighting estimation. In Figure 9, we show results without global lighting, with WSH as global lighting, and with Wsky as global lighting, respectively. We find that P-Net is incapable to learn the correct sun position only with the warped color image but can recover it accurately by adding WSH as shown in the first column and third column. Although the sun position is well recovered with WSH, the sun intensity still has a large gap from the real condition.

For an off-the-shelf renderer (e.g., Blender), we can achieve multi-object rendering by setting it to render only the object in the selected lighting position, and then blending this result with rendering results from other positions through the alpha channel. In Figure 10, we show the visual quality of synthetic object insertion to better illustrate the usefulness of spatially-varying outdoor lighting estimation. As can be observed, our method can render correct lighting effects (specular highlights and shadows) on rabbits under different materials.

### Effects of different losses

To verify the necessity of each loss function, we evaluate the performance of I-Net and P-Net using different combinations of loss functions. In Figure 6, we can observe that our models have comparable improvement to LENetsky even without Ldif due to constraint from intrinsics provided by I-Net. By further adding Ldif, I-Net can learn the global sky environment map more effectively with the guidance of SH encoded lighting and then produce a more accurate sky estimation than LENetsky. If Lcom is ablated, the performance on test dataset becomes 0.760 / 0.564 (SSIM / MAE); by adding this loss, the numbers are 0.798 / 0.552 (SSIM / MAE), which shows that P-Net can predict local lighting more accurately especially on structure similarity.

### 5.2. Evaluation on Real Dataset

#### Real data capture

To validate that SOLID-Net is able to perform outdoor local lighting estimation, we capture real outdoor city street view scenes and the corresponding spatially-varying local environment maps (see Figure 11). The images are captured by a Ricoh Theta SC2 camera with dual fisheye lens. For local lighting environment maps, the scenes are captured 1/2500s shutter speed with f2.0 aperture by placing the panoramic camera as a light probe at different locations. Due to the limited dynamic range of our panoramic camera, the local environment maps are not able to faithfully record the intensity of sunlight. To obtain the accurate sun position for evaluation, we further capture a low-exposure panorama with 1/25000s shutter speed and label the sun position manually. The captured LDR local lighting is aligned to its view vector with respect to the camera facing direction. In total, our real test dataset includes 29 outdoor scenes and 67 LDR local lighting environment maps for evaluating our method quantitatively.

#### Comparison with previous work

We first compare the accuracy of global lighting estimation with LENetsky model [10] using sun position errors. The azimuth/elevation angular errors of LENetsky are 37.5°/9.5°. In contrast,
Figure 11: Comparison of estimated global and local lighting on our real test dataset. Column 1 shows the input image and selected pixel locations. Column 2 shows the estimated global lighting of DeepSky model [10] and our method (blue star marks the estimated sun position by computing the centroid of largest connected component, while the green star marks the ground truth sun position labeled from a low-exposure environment map by us manually). Column 3 shows the LDR local lighting environment map. Columns 4–7 show the estimated local lighting by NeurIllum [20] and our methods in both LDR and HDR formats.

Figure 12: Qualitative comparison of relighting results using our real dataset.

Our method maintains a high accuracy with $28.9^\circ / 6.8^\circ$. From Column 2 of Figure 11, we can see that our method can generate a clearer environment map with different sky conditions and our estimated sun positions are closer to the ground truth than LENet sky. To evaluate the estimated local lighting, we compare our method with NeurIllum [20] retrained on our synthetic dataset on estimated spatially-varying lighting quantitatively and qualitatively. Overall, our method achieves a better SSIM / MAE (the higher is better / the lower is better) performance of $0.235 / 0.203$, compared to $0.228 / 0.233$ for NeurIllum. Compared estimations of our method (Column 4-5) and NeurIllum (Column 6-7) with ground truth (Column 3) in Figure 11, we note that their method does not capture the accurate sun position and intensity, caused by missing panoramic information which our method handles well. We also show relighted bunny results to further compare estimated spatially-varying lighting effects of our method and NeurIllum (see Figure 12). These show that our approach adapts to strongly spatially-varying local lighting effects in real scenes.

6. Discussion

We present the first end-to-end outdoor spatially-varying lighting estimation framework and demonstrate it significantly outperforms previous works via extensive evaluations on both synthetic and real datasets. Our method is able to generalize on real scenes with a slightly different appearance from our synthetic scene. An example is shown in Figure 13, in which the virtual object is reasonably relit in a scene of rarely seen structures (with railway and glass) in the synthetic training data.

Limitations and future work. Due to the material diversity gap between synthetic and real data, the intrinsic decomposition results on real data may not be as accurate as those on synthetic data (Figure 14 compared with Figure 2). Although SOLID-Net estimates HDR lighting environment map to support realistic relighting effects, our lighting model is not suitable for generating animations that are sensitive at harsh lighting boundaries, which will be an interesting direction for future work.
References

[1] Blender. https://www.blender.org.
[2] Blender SceneCity. https://www.cgchan.com/store/scenecity.
[3] HDRHaven. https://hdrhaven.com.
[4] Chang Angel, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. International Conference on 3D Vision (3DV), 2017.
[5] Jonathan T. Barron and Jitendra Malik. Intrinsic Scene Properties from a Single RGB-D Image. In Proc. of Computer Vision and Pattern Recognition, 2013.
[6] Dachuan Cheng, Xiaoming Deng Jian Shi, Yanyun Chen, and Xiaopeng Zhang. Learning Scene Illumination by Pairwise Photos from Rear and Front Mobile Cameras. Computer Graphics Forum, 37:213–221, 2018.
[7] Paul Debevec. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In Proc. of ACM SIGGRAPH, 1998.
[8] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to Predict Indoor Illumination from a Single Image. In Proc. of ACM SIGGRAPH Asia, 2017.
[9] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast Spatially-Varying Indoor Lighting Estimation. In Proc. of Computer Vision and Pattern Recognition, 2019.
[10] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep Sky Modeling for Single Image Outdoor Lighting Estimation. In Proc. of Computer Vision and Pattern Recognition, 2019.
[11] Yannick Hold-Geoffroy, Emiliano Gambaretto Kalyan Sunkavalli, Sunil Hadap, and Jean-François Lalonde. Deep Outdoor Illumination Estimation. In Proc. of Computer Vision and Pattern Recognition, 2017.
[12] Alexei A. Efros Jean-Franc¸ois Lalonde and Srinivasa G. Narasimhan. Estimating the Natural Illumination Conditions from a Single Outdoor Image. International Journal of Computer Vision, 98(2):123–145, 2012.
[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.
[14] JF Lalonde, LP Asselin, J Becirovski, Y Hold-Geoffroy, MA Gardner, and J Zhang. The Laval HDR sky database, 2016. http://sky.hdrdb.com.
[15] Chloe LeGendre, Graham Fyffe Wan-Chun Ma, John Flynn Laurent Charbonnel, Jay Busch, and Paul Debevec. DeepLight: Learning Illumination for Unconstrained Mobile Mixed Reality. In Proc. of Computer Vision and Pattern Recognition, 2019.
[16] Zhengjin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and ssvbdf from a single image. In Proc. of Computer Vision and Pattern Recognition, 2020.
[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In Proc. of Neural Information Processing Systems, 2019.
[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention, 2015.
[19] Ramprasaath R. Selvaraju, Abhishek Das Michael Cogswell, Devi Parikh Ramakrishna Vedantam, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In Proc. of International Conference on Computer Vision, 2017.
[20] Shuran Song and Thomas Funkhouser. Neural Illumination: Lighting Prediction for Indoor Environments. In Proc. of Computer Vision and Pattern Recognition, 2019.
[21] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion from a Single Depth Image. In Proc. of Computer Vision and Pattern Recognition, 2017.
[22] Shuran Song, Andy Zeng, Angel X Chang, Manolis Savva, Silvio Savarese, and Thomas Funkhouser. Im2Pano3D: Extrapolating 360 Structure and Semantics Beyond the Field of View. In Proc. of Computer Vision and Pattern Recognition, 2019.
[23] Jessi Stumpfel, Andreas Wenger Andrew Jones, Tim Hawkins Chris Tchou, and Paul Debevec. Direct HDR Capture of the Sun and Sky. In Proc. of ACM SIGGRAPH, 2004.
[24] Ye Yu and William AP Smith. InverseRenderNet: Learning single image inverse rendering. In Proc. of Computer Vision and Pattern Recognition, 2019.
[25] Amir Roshan Zamir and Mubarak Shah. Image Geolocalization Based on Multiple Nearest Neighbor Feature Matching using Generalized Graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, PP(99):1–1, 2014.
[26] Jinsong Zhang, Yannick Hold-Geoffroy Kalyan Sunkavalli, Jonathan Eisenmann Sunil Hadap, and Jean-François Lalonde. All-Weather Deep Outdoor Lighting Estimation. In Proc. of Computer Vision and Pattern Recognition, 2019.
[27] Jinsong Zhang and Jean-François Lalonde. Learning High Dynamic Range from Outdoor Panoramas. In Proc. of International Conference on Computer Vision, 2017.
[28] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, and Thomas Funkhouser. Physically-Based Rendering for Indoor Scene Understanding Using Convolutional Neural Networks. In Proc. of Computer Vision and Pattern Recognition, 2017.
In this supplementary material, we provide more details about our implementation details, synthetic data generation, network architectures, and additional results on synthetic data and real images in the wild.

1. Implementation Details

To train SOLID-Net, we use SOLID-Img augmented with random flip and crop. Our framework is implemented in PyTorch [17] and Adam [13] optimizer is used with default parameters. We first train I-Net using a batch size of 8 for 20 epochs until convergence, and then train P-Net with a batch size of 4 for 60 epochs on an RTX2080 GPU. We find that an end-to-end fine-tuning does not improve the performance. The learning rate is initially set to $5 \times 10^{-4}$ and halved every 5 epochs for both networks. Training convergence takes roughly 24 hours.

2. Additional Details in Data Generation

The data generation pipeline for SOLID-Img dataset has been introduced in Section 3.1 of the main paper. Here, we introduce additional details about the camera setting step.

For each road block, we select a set of cameras with diverse views seeing most objects in the context, to provide comprehensive information for lighting estimation, as shown in Figure 2(a) of the main paper. Our process starts by selecting the “best” camera [28] for each of the six horizontal view direction sectors in every road block. For each horizontal view direction, we sample a dense set of cameras on a 2D grid with 1m solution, choosing a random camera viewpoint within each grid cell, a random horizontal view direction within the 60$^\circ$ sector, a random height of $1.55 \pm 0.05$m above the floor uniformly, and a pitch angle within $10^\circ$ around horizontal direction. Then for each camera, we render an item buffer and count the number of visible pixels according to Z-buffer and the number of objects. For each horizontal view direction in each road block, we select the view with the highest percentage pixel coverage, as long as it has more than three object categories.

3. Details about Network Architectures

In this section, we introduce the detailed network architectures of baseline models as shown in Figure 1.

The first baseline model, denoted as LENet$_{reg}$, is a regression model that directly regresses the global sky environment map from the input limited-FOV image. The second baseline model, denoted as LENet$_{ae}$, is a two-stream convolution network used to estimate sun position and normalized HDR panorama from a LDR panorama [27]; we modify the input as a single limited-FOV LDR image to adapt our task. The last baseline model, denoted as LENet$_{sky}$, learns to estimate both the sun azimuth and the sky parameters from two image encoders and uses an autoencoder to learn the space of outdoor lighting by compressing an HDR sky image to a 64-dimensional latent vector and reconstructing it to a HDR sky environment map [10]. In particular, LENet$_{ae}$ learns azimuth estimation as a regression task, while LENet$_{sky}$ treats it as a classification task.

Numerical results about these baseline models compared with ours are shown in Table 2 ($\xi_{azimuth}$ is the sun azimuth angular error, $\xi_{elevation}$ is the sun elevation angular error, and $\xi_{HDR}$ is the MAE error of normalized sky environment map). Training a single model to estimate a sky environment map with azimuth rotation is proved to be difficult (see Row 1 in Table 2). By separating this task into the estimation of a normalized sky environment map (sun in the middle position along the horizontal direction) and sun azimuth angle, the results of single model lighting estimation get improved (see Row 2 in Table 2). Further using three sub models to solve the whole problem results in more accurate estimation than using single models (see Row 3 in Table 2). But the sun position and sky environment map are closely related, we prove that the estimation accuracy to both could be improved by our jointly training with intrinsic constraints being integrated in deep models (see Row 4-5 in Table 2).

4. Qualitative Results on Synthetic Data

More results of image decomposition and lighting estimation using SOLID-Img test dataset are shown in Figure 2, Figure 3, and Figure 4. Given an input image, our estimated albedo, normal, plane distance, shadow, and shading show close appearance to the ground truth (shown as insets) as shown in Figure 2. In Figure 3, we can see that global lighting estimation results of SOILD-Net are closer to ground truth than three baseline models (we rotate the lighting of LENet$_{ae}$ and LENet$_{sky}$ according to the sun azimuth for better comparison). In addition, the relighted bunnies using our estimated lighting display accurate cast shadows, while other models fail to render.

As shown in Figure 4, our method can recover more accurate local lighting than NeurIllum [20] even for the reflection of the ground and some unseen parts (typical examples could be found in local lighting 1 in row 1, local lighting 3

Table 2: Numerical results and MAE errors on estimated sky environment map.

| Methods          | $\xi_{azimuth}$ | $\xi_{elevation}$ | $\xi_{HDR}$ |
|------------------|-----------------|-------------------|-------------|
| LENet$_{reg}$    | 37.0$^\circ$    | 16.2$^\circ$      | 0.508       |
| LENet$_{ae}$ [27]| 34.0$^\circ$    | 16.0$^\circ$      | 0.542       |
| LENet$_{sky}$ [10]| 22.3$^\circ$    | 11.0$^\circ$      | 0.609       |
| Ours (w/o $L_{dit}$) | 19.1$^\circ$ | 11.4$^\circ$ | 0.491       |
| Ours             | 12.6$^\circ$    | 8.5$^\circ$       | 0.478       |
5. Qualitative Results on Real Data

More results of local lighting estimation on real dataset are shown in Figure 5. Compared with NeurIllum [20], the lighting estimation results of our method are more similar to the ground truth in terms of overall structure, and our relighted bunnies show more realistic rendering appearances.

Figure 1: Structures of baseline models.
Figure 2: Intrinsic decomposition results and ground truth (shown as insets) on SOLID-Img test dataset.
| Input Image | \(\text{LEN}_{\text{reg}}\) | \(\text{LEN}_{\text{ae}}\) | \(\text{LEN}_{\text{sky}}\) | Ours | GT |
|-------------|----------------|----------------|----------------|------|----|
| ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) | ![Image](image4.png) | ![Image](image5.png) | ![Image](image6.png) |
| ![Image](image7.png) | ![Image](image8.png) | ![Image](image9.png) | ![Image](image10.png) | ![Image](image11.png) | ![Image](image12.png) |
| ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) | ![Image](image16.png) | ![Image](image17.png) | ![Image](image18.png) |
| ![Image](image19.png) | ![Image](image20.png) | ![Image](image21.png) | ![Image](image22.png) | ![Image](image23.png) | ![Image](image24.png) |
| ![Image](image25.png) | ![Image](image26.png) | ![Image](image27.png) | ![Image](image28.png) | ![Image](image29.png) | ![Image](image30.png) |
| ![Image](image31.png) | ![Image](image32.png) | ![Image](image33.png) | ![Image](image34.png) | ![Image](image35.png) | ![Image](image36.png) |

Figure 3: Global lighting estimation and relighting results on SOLID-Img test dataset.
Figure 4: Local lighting estimation and relighting results on SOLID-Img test dataset.
Figure 5: Spatially-varying lighting estimation and relighting results on real dataset.