UrbanIR: Large-Scale Urban Scene Inverse Rendering from a Single Video

Supplementary Material

Anonymous CVPR submission

Paper ID

1. Method

UrbanIR takes a multi-frame video of a scene under single illumination; as the camera moves, its motion is known. Write $\{I_i, E_i, K_i\}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$ is the RGB image; $E_i \in \mathrm{SE}(3)$ is the camera pose; and K_i is camera intrinsic matrix. We produce a neural field model that can be viewed from *novel camera viewpoints* under *novel lighting conditions*. We do so by constructing a neural scene model that encodes albedo, normal, semantics, and visibility in a unified manner (Sec. 1.1). This model is rendered from a given camera pose with given illumination using an end-to-end differentiable volume renderer (Sec. 1.2). Our inference is by joint optimization of all properties (Sec. 1.3). Applications include changing the sun angle (**??**; top right), day-to-night transitions (**??**; bottom right), and object insertion (**??**; middle right). More details about applications are in Sec. 2. Fig. **??** provides an overview of our proposed inverse graphics and simulation framework.

1.1. Relightable Neural Scene Model

The scene representation is built on Instant-NGP [53, 58], a spatial hash-based voxel NeRF representation. Instant-NGP offers numerous advantages, including low memory consumption; high efficiency in training and rendering; and compatibility with expansive outdoor scenes. Write $\mathbf{x} \in \mathbb{R}^3$ for position in 3D, $\mathbf{d}$ for query ray direction, θ for learnable scene parameters; NeRF models, including Instant-NGP, learn a radiance field $F_\theta(\mathbf{x}, \mathbf{d}) = (\mathbf{c}, \sigma)$, where $\mathbf{c} \in \mathbb{R}^3$ and $\sigma \in \mathbb{R}$ represent observed color and opacity respectively. Standard NeRFs have view- and lighting-dependent effects, such as shading, shadow, and specularity, baked into their observed color, making them non-relightable.

In contrast, UrbanIR learns a model of the intrinsic scene attributes field independent of viewing angles and lighting conditions. Write diffuse albedo $\mathbf{a}$, surface normal $\mathbf{n}$, semantic vector $\mathbf{s}$, and density σ; then UrbanIR learns:

$$F_\theta(\mathbf{x}) = (\mathbf{a}, \mathbf{n}, \mathbf{s}, \sigma) \tag{1}$$

where θ is learnable parameters. The diffuse albedo represents the intrinsic color and texture of the material; the normal represents the intrinsic surface geometry; density encodes the spatial opacity, and semantics is used as a key to query surface reflectance. Following Instant-NGP [53], we learn a dense feature hash table to represent the scene, and an individual MLP header is used to decode each attribute given a queried feature at point $\mathbf{x}$. We provide the details of the architecture in the supplementary. The geometry of the scene is implicitly encoded in σ. In contrast to existing relightable outdoor scene models that demand coupled explicit geometry [60, 73], our scene model is implicit, providing compactness and consistency to appearance modeling.

The lighting model is a parametric sun-sky model [34, 80]. This encodes outdoor illumination as:

$$\mathbf{L} = \{(\mathbf{L}_{\text{sun}}, \psi_{\text{sun}}, \phi_{\text{sun}}), \mathbf{L}_{\text{amb}}, \mathbf{L}_{\text{sky}}\}. \tag{2}$$

Our sun model is a 5-DoF representation, encoding sun color $\mathbf{L}_{\text{sun}}$ along with the azimuth and zenith $\psi_{\text{sun}}, \phi_{\text{sun}}$. The $\mathbf{L}_{\text{amb}}$ model is represented as a 3-DoF ambient light. The sky dome model infers the sky texture from the viewing direction: $\mathbf{C}_{\text{sky}} = \mathbf{L}_{\text{sky}}(\mathbf{r})$. We chose this minimalist sun-sky model as it is more compact than other alternatives (e.g., HDR dome or Spherical Gaussians) yet has proven highly effective in modeling various outdoor illumination effects [34, 80].

1.2. Rendering

Given the scene model F_θ and a lighting model $\mathbf{L}$, rendering involves two steps: 1) volume rendering of the scene's intrinsic properties and visibility map onto the image plane, and 2) a shading process to produce the final result with view-dependent and lighting-dependent effects:

$$\mathbf{C} = \texttt{Shade}(\texttt{Intrinsic}(F_\theta, \mathbf{r}), \texttt{Shadow}(F_\theta, \mathbf{r}, \mathbf{L}), \mathbf{L}) \tag{3}$$

where $\mathbf{L}$ is the lighting model, $\mathbf{C}$ is the final RGB color.

Intrinsics images are obtained by volume rendering. We accumulate predictions from $F(\cdot;\theta)$ along the query ray. Multiple points are sampled along the ray, and intrinsics

at the query pixel along the ray [29, 51]. In particular, the albedo $\mathbf{A}$, normal $\mathbf{N}$, and semantics $\mathbf{S}$ are predicted as:

$$\mathbf{A}(\mathbf{r})=\sum_{i=1}^{N} w_i \mathbf{a}_i, \mathbf{N}(\mathbf{r})=\sum_{i=1}^{N} w_i \mathbf{n}_i, \mathbf{S}(\mathbf{r})=\sum_{i=1}^{N} w_i \mathbf{s}_i, \quad (4)$$

where $w_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)\,(1-\exp(-\sigma_i \delta_i))$ is alpha-composition weight, $\delta_i = t_i - t_{i-1}$. We perform rendering for each camera ray and get the final semantic map, albedo map, and the normal map.

Shadow modeling and rendering are essential for obtaining realistic-looking outdoor images. Modeling the visibility of the sun with a per-scene optimized MLP head (as in [85, 87]) is impractical because we need to change the sun's position in relighting but can learn from only one position. An alternative is to construct an explicit geometry model to cast shadows [73], but this model might not be consistent with the other neural fields, and imposing consistency is difficult. Instead, we first compute an estimate $\mathbf{x}(\mathbf{r})$ of the 3D point being shaded, then estimate the visibility $V(\mathbf{x}, \text{sun})$. Our key insight is that shadows in outdoor scenes are primarily due to the visibility of a single directional sunlight.

We obtain $\mathbf{x}(\mathbf{r})$ for each ray by volume rendering depth (so substitute $\hat{t} = \sum w_i t_i$ into the equation for the ray being rendered). Now, to check whether $\mathbf{x}$ is visible to the light source, we compute the transmittance along the ray segment between $\mathbf{x}$ and the light source using volume rendering:

$$V(\mathbf{x}, \text{sun})=\exp\left(-\sum_i \sigma_i(\mathbf{x}_i)\delta_i\right) \text{ where } \mathbf{x}_i = \mathbf{x} + t_i \mathbf{l}_{\text{sun}} \quad (5)$$

Lower transmittance along a ray from a surface point to a light source suggests fewer obstacles between the point and the light source. Eq. 5 establishes a strong link between transmittance, lighting, and visibility fields used in training. In particular, a point in a training image known as shadowed (resp. out of shadow) should have large (resp. small) accumulated transmittance. We use this constraint to adjust distant geometry during training. Compared to other alternatives [73, 87], our proposed visibility test is simple to compute, flexible for relighting, and aligns with intrinsic properties with a few mild assumptions for outdoor scenes.

Shading is performed by a Blinn-Phong model [7] that incorporates sun and sky terms for the foreground scene and an MLP query for the background sky. For $\mathbf{S}(\mathbf{r}) \in$ sky, we use $\mathbf{C}(\mathbf{r}) = \mathbf{L}_{\text{sky}}(\mathbf{r})$ and otherwise, we use

$$\mathbf{C}(\mathbf{r}) = \mathbf{A}(\mathbf{r})\,(\mathbf{L}_{\text{sun}}\mathbf{D}\mathbf{V} + \mathbf{L}_{\text{amb}}) \quad (6)$$

where $\mathbf{D} = \max(\mathbf{N}(\mathbf{r}) \cdot \mathbf{l}_{\text{sun}}, 0)$ is the diffuse lighting at the surface, $\mathbf{l}_{\text{sun}}$ is the sunlight direction (derived from $\psi_{\text{sun}}, \phi_{\text{sun}}$). The visibility $V(\mathbf{x}, \text{sun})$ is 1 if $\mathbf{x}(\mathbf{r})$ can see the sun and 0 otherwise. This shading model is capable of producing a realistic appearance with shadows following varying lighting conditions. The model can readily be extended with additional lighting sources at the relighting stage, as later shown in the night simulation.

1.3. Inverse graphics

We train scene $F(\cdot)$ (Eq. 1) and lighting $\mathbf{L}$ (Eq. 2) models jointly using a loss:

$$\min_{\theta,\mathbf{L}} \mathcal{L}_{\text{render}}+\lambda_0\mathcal{L}_{\text{deshadow}}+\lambda_1\mathcal{L}_{\text{visibility}}+\lambda_2\mathcal{L}_{\text{normal}}+\lambda_3\mathcal{L}_{\text{semantics}}+\lambda_4\mathcal{L}_{\text{reg}}, \quad (7)$$

where individual loss terms are described below.

Rendering loss measures the agreement between observed images and images rendered from the model using the training view and lighting, yielding $\mathcal{L}_{\text{render}} = \sum_{\mathbf{r}} \|\mathbf{C}_{\text{gt}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2$, where $\mathbf{C}$ is rendered color per ray, as defined in Eq. 3, and $\mathbf{C}_{\text{gt}}$ is the observed "ground-truth" color. Minimizing the rendering loss ensures our scene model can reproduce the observed images.

Deshadowed rendering loss forces shadow effects out of the estimated albedo. In particular, we compute a shadow-free version of an image using an off-the-shelf shadow detection and removal network [13, 22] to obtain $\mathbf{C}_{\text{deshadow}}$. We then render that image from the model using the training view and lighting, but assuming that every point can see the sun (equivalently $V(\mathbf{x}, \text{sun}) = 1$ for every $\mathbf{x}$). This yields $\mathbf{C}'(\theta)$. We then measure the agreement between the two to obtain $\mathcal{L}_{\text{deshadow}} = \sum_{\mathbf{r}} |\mathbf{C}_{\text{deshadow}} - \mathbf{C}'(\theta)|^2$. The combination of this deshadowed rendering loss and the original rendering loss directly gauges how the visibility map influences rendering, and helps disentangle albedo and shadows.

Visibility loss exploits shadow detection to improve geometry estimates. A pixel that is known to be in shadow must be at a point that *cannot see the sun*, so constraining geometry along a ray from that pixel to the sun. This loss could be computed by simply comparing visibility $V(, \text{sun})$ with the shadow masks used for $\mathcal{L}_{\text{deshadow}}$. However, there are challenges: first, computing visibility requires another volume rendering per sample point; second, back-propagation through volume rendering, shading, and visibility computation forms a long, non-linear gradient chain, and optimization becomes difficult. Instead, we construct an intermediate "guidance" visibility estimate $\hat{V}(\mathbf{r})$ which is an MLP head trained to reproduce the shadow masks, and compute

$$\mathcal{L}_{\text{visibility}} = \sum_{\mathbf{r}\in\mathcal{R}} \text{CE}\left(M(\mathbf{r}), \hat{V}(\mathbf{r})\right) + \text{CE}\left(V(\mathbf{r}), \hat{V}(\mathbf{r})\right), \quad (8)$$

where $M(\mathbf{r})$ is the shadow mask at pixel $\mathbf{r}$, , and CE(., .) is a cross-entropy loss. Here the first term forces the (relatively easily trained) $\hat{V}$ to agree with the shadow masks, and the second forces V to agree with $\hat{V}$.

Figure 1. **Night-time rendering.** In this sequence of images, the scene changes from daytime to night-time by introducing new light sources such as a headlight on a car and a street lamp. The top three and bottom three rows are from the same driving video, but at different times. Our decomposition method successfully removes dark shadows with sharp boundaries, resulting in a more realistic rendering of new light sources (such as streetlights and headlights) during night-time simulations. Our method is superior to Instruct-NeRF2NeRF [23], a data-driven, generative prior, radiance field approach.

Normal loss is computed by comparing results N_{gt} from an off-the-shelf normal estimator [18, 30] to the output of the normal MLP. Recall the camera is known for training scenes and write $\mathbf{r}$ for the pixel corresponding to 3D point $\mathbf{x}(\mathbf{r})$. An alternate estimate of the normal follows from the density field: $\hat{N}(\mathbf{r}) = -\frac{\nabla\sigma(\mathbf{x})}{\|\nabla\sigma(\mathbf{x})\|}$. Then our normal loss is given by:

$$\mathcal{L}_{\text{normal}} = \sum_{\mathbf{r}\in\mathcal{R}} \left(\|N_{\text{gt}}(\mathbf{r}) - N(\mathbf{r})\|^2 + \|N(\mathbf{r}) - \hat{N}(\mathbf{r})\|^2 \right). \tag{9}$$

We adopt smooth normal regularization from Ref-NeRF [69], producing better density. The normal loss requires second-order derivatives of the density during back-propagation. An efficient implementation using the Hessian vector products (HVP) enables memory-efficient computation.

Semantic loss is computed by comparing predicted semantics $\mathbf{s}$ with labels in the dataset [42]. We use an additional loss to encourage high-depth values in the sky region, yielding: $\mathcal{L}_{\text{semantics}} = \sum_{\mathbf{r}\in\mathcal{R}} \text{CE}\left(S_{\text{gt}}(\mathbf{r}), S(\mathbf{r})\right) - \sum_{\mathbf{r}\in\text{sky}} D(\mathbf{r})$.

A regularization term is used to regularize the albedo of the scene and ambient light intensity. This is necessary due to the ill-posed nature of our optimization process. However, removing the hard shadow from the sunlight in the albedo field $\mathbf{A}$ remains a challenge, particularly in urban driving sequences. To address this challenge, we introduce a prior that ensures the ground albedo is homogeneous. This is important because the ground region typically shares a similar albedo value. More specifically, we first compute the average ground albedo $\bar{\mathbf{A}}_{\text{g}}$ from albedo $\mathbf{A}$ and semantic S_{gt} and regularize the albedo using $\mathcal{L}_{\text{albedo}} = \sum_{\mathbf{r}\in\text{ground}} \|\mathbf{A}(\mathbf{r}) - \bar{\mathbf{A}}_{\text{g}}\|_2$.

We also calculate an *ambient regularization* term as $\|\mathbf{L}_{\text{amb}}\|_2$. We regularize the intensity of ambient light to avoid unnatural color shifts in the recovered albedo caused by a large intensity of ambient light. Our regularization term is thus $\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{albedo}} + \|\mathbf{L}_{\text{amb}}\|_2$.

2. Application Details

As intrinsics are recovered, UrbanIR can be rendered using any preferred source model. Natural uses are rendering scenes with different sun configurations and simulating night-time.

Simulating night-time proceeds by defining headlights and street lights, then illuminating with scene model considering specularity and lens flare. For sky regions $\mathbf{S}(\mathbf{r}) \in$ sky, we use $\mathbf{C}(\mathbf{r}) = \mathbf{L}_{\text{sky}}(\mathbf{r})$ and otherwise, we use

$$\mathbf{A}(\mathbf{r}) \left(\sum \mathbf{L}^{\text{i}}_{\text{dif}} \mathbf{D}_i \mathbf{V}_i + \mathbf{L}_{\text{amb}} \right) + \sum_i \mathbf{L}^i_{\text{spec}} \tag{10}$$

The spotlight we used is given by the center $\mathbf{o}_L^i \in \mathbb{R}^3$ and direction $\mathbf{d}_L^i \in \mathbb{R}^3$ of the light. This spotlight produces a diffuse radiance at $\mathbf{r}$ given by

$$\mathbf{L}_{\text{dif}}^i(\mathbf{r}) = \frac{1}{\|\mathbf{o}_L^i - \mathbf{x}(\mathbf{r})\|^2} \left(l \cdot \mathbf{d}_L^i\right)^k, l = \frac{\mathbf{o}_L^i - \mathbf{x}(\mathbf{r})}{\|\mathbf{o}_L^i - \mathbf{x}(\mathbf{r})\|}, \tag{11}$$

Spotlight's diffuse color intensity is brightest on the central ray $\mathbf{r}(t) = \mathbf{o}_L - t\mathbf{d}_L$, decays with distance from ray $\mathbf{r}(t)$ and angle. We modulate it with constant k.

The realistic night-time simulation requires reproducing the strong specular effects on cars. We find car regions using a semantic field $\mathbf{S}$ in Eq. 4, then simulate specular reflection with the Blinn-Phong model [7], where the γ (specular strength) parameter is inherited from the semantic field.

At night, luminaires often display lens flares. A pure simulation of lens flares is impractical, as it requires extensive ray tracing through the lens. We use the standard image-based approximation [1] to simulate such light scattering effects. For directly visible luminaires, we composite a real-world lens flare image from a similar lighting source into the image, using location and depth. As Fig. 1 shows, this simple method is effective.

Object insertion proceeds by a hybrid rendering strategy. We first cast rays from the camera and estimate ray-mesh intersections [16] for the inserted object. If the ray hits the mesh and the distance is shorter than the volume rendering depth, the albedo $A(\mathbf{r})$, normal $N(\mathbf{r})$, and depth $D(\mathbf{r})$ are replaced with the object attributes. In the shadow pass, we calculate visibility from surface points to the light source (Eq. 5), and also estimate the ray-mesh intersection for the tracing rays. If the rays hit the mesh (meaning occlusion by the object), the visibility is also updated : $V(\mathbf{r}) = 0$. With updated $A(\mathbf{r}), N(\mathbf{r}), V(\mathbf{r})$, shading (Eq. 1.2) is applied to render images with virtual objects. Our method not only casts object shadows in the scene but also casts *scene shadows* on the object, enhancing realism significantly. Similar approaches have been depicted in recent works [37, 57]. However, ours is the first to be visibility-aware, enabling us to render effects when an object enters into a shadow.

Outdoor relighting is done by simply adjusting lighting parameters (position or color of the sun; sky color) then re-rendering using Eq. 3. We also use semantics to interpret specular car surfaces and emulate their reflectance during the simulation.

3. Model Architecture

Instant-NGP [53] encodes the scene with a multi-scale hash table, and each entry contains learnable parameters. For point $\mathbf{x} \in \mathbb{R}^3$, the model retrieves and interpolates the parameters with hash function: $F(\mathbf{x}, \theta)$. UrbanIR adopts the hash encoding from [53] and maintain two separate hash tables for geometry and appearance, and predict the scene properties with:

$$\begin{aligned} \sigma &= F_g(\mathbf{x}, \theta_g) \\ (\mathbf{a}, \mathbf{n}, s) &= F_a(\mathbf{x}, \theta_a), \end{aligned} \tag{12}$$

where σ is density, $(\mathbf{a}, \mathbf{n}, s)$ are albedo, surface normal, and semantic. θ_g, θ_a are learnable parameters for geometry and appearance. Please note that the density field σ is not only involved in the volume rendering (Eq. 4), but also involved in visibility estimation (Eq. 5) and normal loss calculation (Eq. 9). The hash encoding is implemented with tiny-cuda-nn [52]. We empirically find that maintaining separate learnable parameters for geometry and appearance leads to more stable convergence and higher rendering quality.

4. Training Details

The training procedure is illustrated in Fig. 2. We leverage pretrained networks as 2D priors during training to address the ill-posed inverse problem. Specifically, the shadow mask is estimated with MTMT [13], and shadow removal is performed with ShadowFormer [22]. Omnidata normal estimation [18] helps refine scene geometry (Eq. 9), which is critical in the shading quality and albedo decomposition. A semantic map is provided in Kitti360 dataset [42] and can also be estimated with MMSegmentation [15] if such information is not provided. The loss terms are weighted during training:

$$\mathcal{L} = \mathcal{L}_{\text{render}} + \lambda_0 \mathcal{L}_{\text{deshadow}} + \lambda_1 \mathcal{L}_{\text{visibility}} + \lambda_2 \mathcal{L}_{\text{normal}} + \lambda_3 \mathcal{L}_{\text{semantics}} + \lambda_4 \mathcal{L}_{\text{reg}},$$

where $\lambda_0 = 1.0, \lambda_1 = 0.001, \lambda_2 = 0.01, \lambda_3 = 0.04, \lambda_4 = 0.1$. We use Adam optimizer [31] with a learning rate of 0.002 for a total of 100 epochs during the optimization.

5. Related Works Comparison

We compare the problem setting, input requirement with recent methods in Tab. 1. UrbanIR addresses inverse rendering for large-scale urban scenes that object-centric methods [28, 85, 87] fails to reconstruct. Furthermore, our method takes videos under single illuminations as input. In order to estimate the geometry of large-scale driving scenes, FEGR [73] and LightSim [56] rely on captures from five to six cameras and LiDAR sensors. On the other hand, UrbanIR only needs videos from single or stereo cameras without any guidance from LiDAR. Our method also performs nighttime simulation by inserting local light sources (e.g. streetlight, vehicle light), which is not demonstrated in previous works.

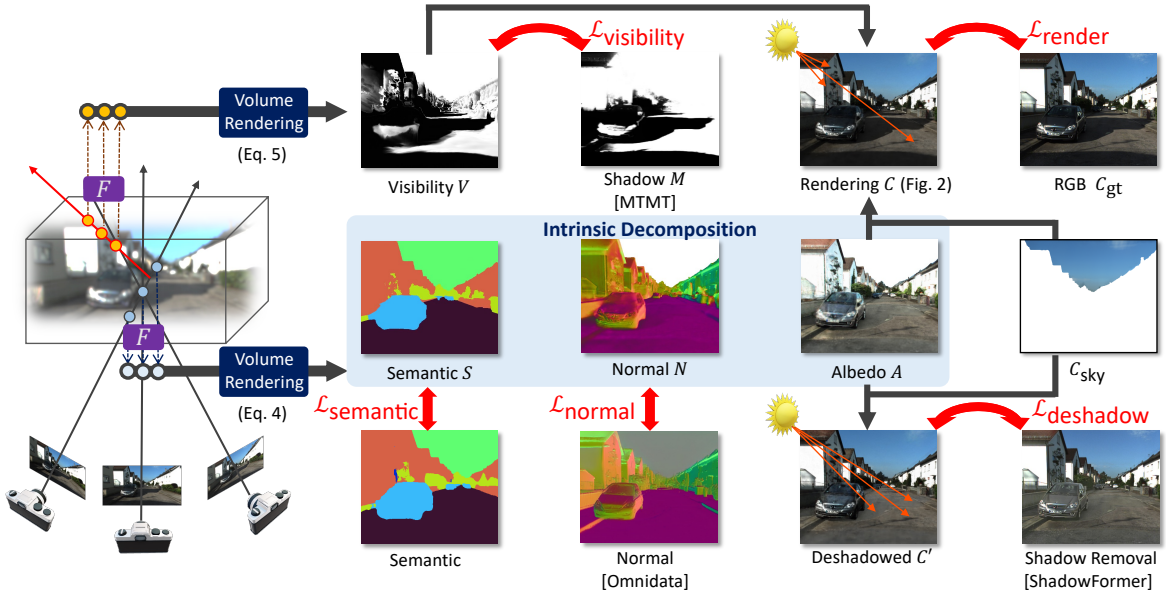

Figure 2. **Training Pipeline.** UrbanIR retrieves scene intrinsics with volume rendering from camera rays, which is guided by semantic and normal priors. Transmittance along tracing rays is supervised with shadow masks. The shading model (illustrated in Fig. **??**) is performed **with** and **without** visibility term, and enforce reconstruction loss with original and deshadowed images, respectively. Please refer to Section 1.3 for more details.

6. Visibility Modeling for Object Insertion

Following [72, 73], we build the object insertion pipeline with Blender [14], and the results are shown in Fig. 3. By tracing the rays from object surface toward light sources (i.e. the sun), UrbanIR estimates the visibility with volume rendering (Eq. 5 in main paper). As a result, our full model is able to cast scene shadow on the inserted objects and also weaken the object shadow on the ground if it overlaps with the existing scene shadow. The visibility modeling enables our method to simulate shadows better and to enhance the insertion realism significantly.

7. More Relighting Results

We compare the relighting quality with FEGR [73] in Fig. 4. FEGR [73] first extracts mesh and estimates the shading from the lighting configuration, and the imperfect mesh geometry produces artifacts and loses appearance details. On the other hand, our method alleviates the original shadow and produces relighting images while preserving appearance details. Additionally, UrbanIR can insert local light sources and simulate the scene in the night time (Fig. 5), demonstrating the capability and flexibility of our relighting framework.

We show additional night simulation results on various Kitti360 [42] sequences in Fig. 6, demonstrating the generalization capability of UrbanIR . The Instruct-Pix2Pix [10] leverages the large language model [11] and stable diffusion [59] for abundant image editing tasks. However, such a data-driven method cannot move the daylight shading and shadow in the input images. On the contrary, UrbanIR decomposes shadow-free albedo and performs physically-based rendering with new light sources (e.g., streetlights, headlights), significantly enhancing the visual quality of night simulation. The strong specular reflection is also simulated on the car region (Eq. 10), boosting the realism of metal material. Please note that the simulation is flexible, and the user can adjust physical parameters (e.g., light color, light strength) to create various effects. Please refer to our supplementary video to visualize view consistency and controllable simulation better.

8. Baseline Details

Description of the approach of baselines we compared to.

NeRF + Mesh The recent work FEGR [73] explores the relighting of outdoor scenes under singular or multiple illumination sources. However, due to the absence of open-source access to their method, we implement our baseline model incorporating similar visibility modeling strategies. Specifically, we employ the marching cubes technique [48] to extract the mesh from our model, excluding our proposed visibility optimization (as per Eq.8). In alignment with the shadow mapping approach adopted by FEGR [73], we cast shadows by estimating two intersections: the first between the camera rays and the mesh and the second by tracing rays

Figure 3. **Object Insertion Qualitative Results.**Without visibility modeling (middle column), the scenes do not cast shadows on the inserted objects and the original object shadow looks unrealistic in the existing shadow. Our full method (right column) simulates the better interaction between the reconstructed scenes and inserted objects with the help of visibility modeling.

from the surface to the light source.

Colmap MVS [64] We compare our method with an explicit geometry-based baseline. For this, we utilize COLMAP for dense scene reconstruction [63, 64] and import the resulting scene into Blender [14] for relighting simulation.

Instruct-Pix2Pix [10] edits images according to user instruction. The model leverages large language model GPT-3 [11] and Stable Diffusion [59] for generating image and instruction pairs and fine-tune diffusion model to perform editing. We use instructions "change to night", and "It's now midnight" for night image generation.

Instruct-NeRF2NeRF [23] aims to edit NeRF scenes with text instructions. It uses a generative image editing model [10] to iteratively edit input images while optimizing the underlying scene model, resulting in an optimized 3D scene that respects the instruction. We compare Instruct NeRF2NeRF in night simulation, where we provide the instruction, "*Make it look like it was taken at night*."

NeRF-OSR [60] is a recent work for outdoor scene reconstruction and relighting. We use the open-source project provided by the author to run this baseline. This method represents lighting as spherical harmonics parameters. It is worth noting that NeRF-OSR was designed for inverse rendering in *multi-illumination conditions*. For a fair comparison, we rotate the spherical vectors to simulate different light conditions.

RelightNet [79] is a single-image based relighting framework. We use the open-source project provided by the authors to produce intrinsic decomposition results, including shading and albedo for comparison.

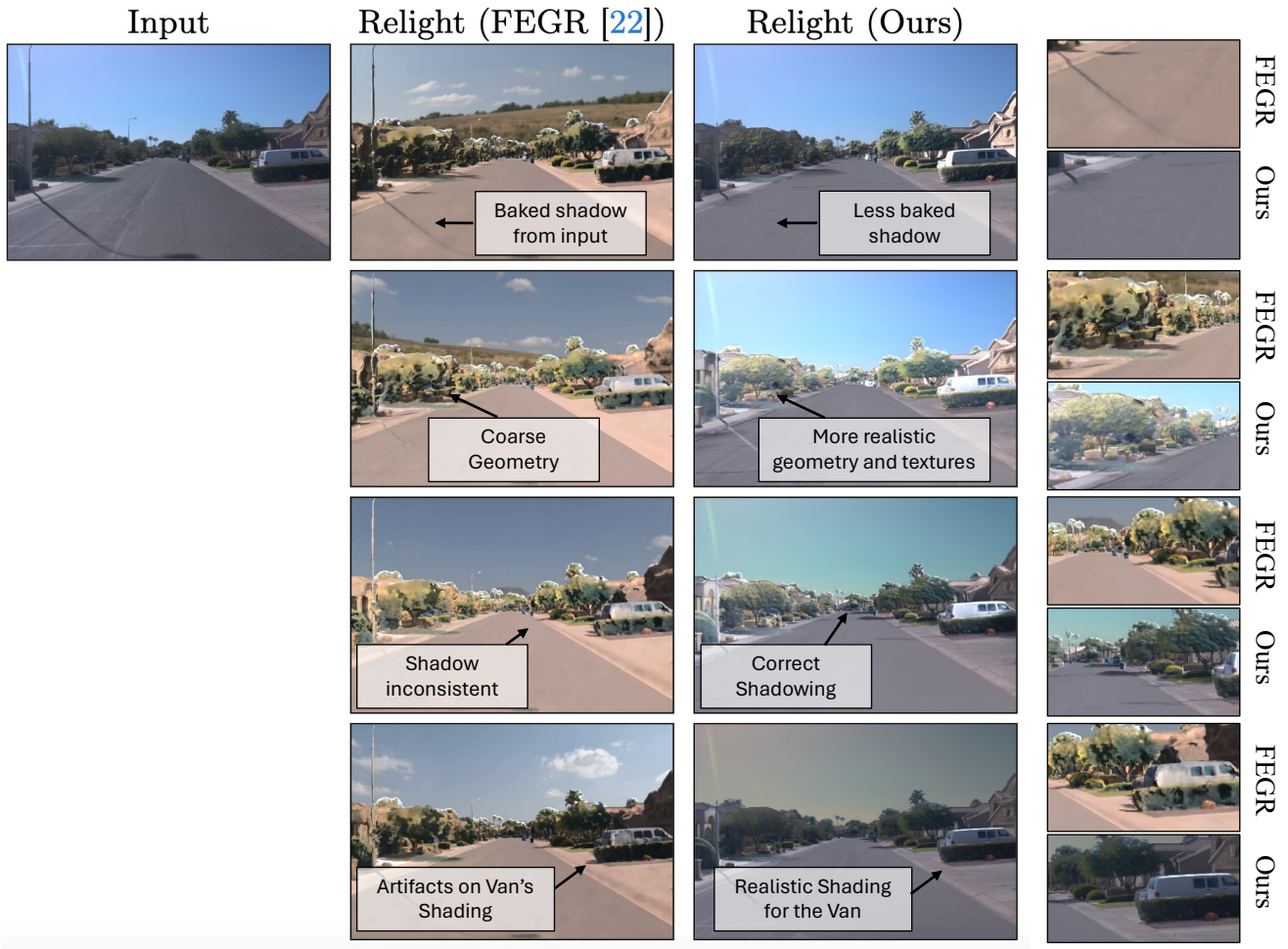

Figure 4. **Relighting Comparison on Waymo Open Dataset [68]. The second and third columns compare the relighting quality. The authors provide the FEGR results and we match the lighting condition according to the shadow direction.**

Figure 5. **Controllable Night Simulation.**

9. Related Work

Inverse Graphics involves inferring illumination and intrinsic properties of a scene. The problem is underconstrained, and there is much reliance on priors [2, 3, 26, 27, 35, 50, 62, 78, 83] or on managed lighting conditions [2, 2, 20, 25, 25, 82], known geometry [17, 32, 36, 61], or material simplifications [49, 83, 88]. Recent methods use deep learning techniques to reason about material properties [45–47, 55, 77, 86]. Models trained on synthetic data [43] or pair-wise annotated data [4] have shown promising results. Learned predictors of albedo or shading are described and reviewed in [6, 19, 65]. Neural representations of material or illumination appear in [5, 38–41, 47]. Like these methods, we exploit monocular cues, such as shadows and surface normals. In contrast, we combine learning-based monocular cues and model-based relightable NeRF optimization to infer the scene's intrinsic properties and illumination.

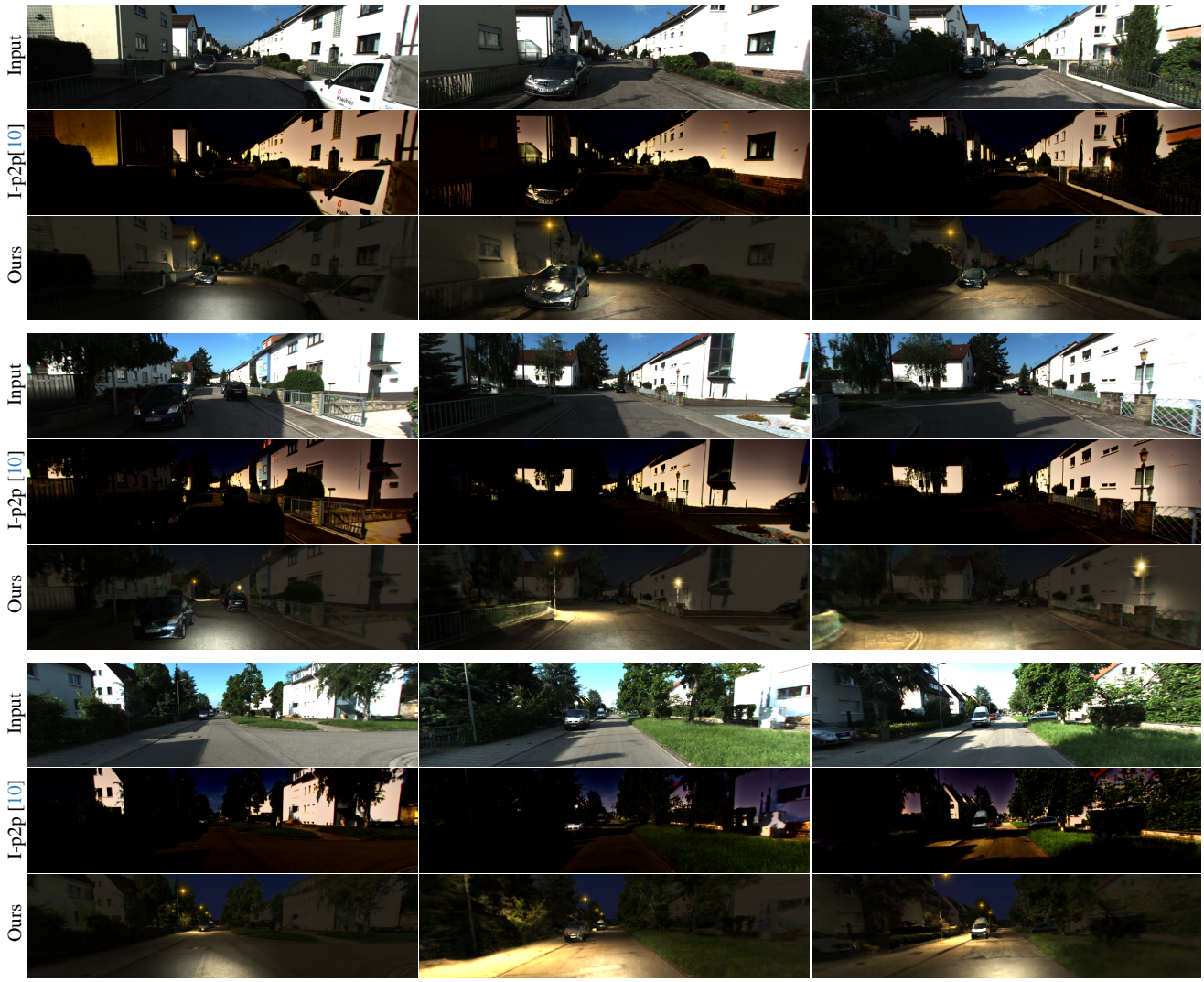

Figure 6. **Nighttime rendering.** The scene is transformed from daytime (1st row) to night-time (3rd row) by introducing new light sources: a headlight on a car and a street lamp. Top 3 and bottom 3 rows are from same driving sequence with different time stamp. Comparing with data-driven generative model and Instruct-Pix2Pix [10], the dark shadows with sharp boundaries are successfully removed with our decomposition, resulting more realistic rendering with new light sources (e.g. streetlights, headlight) during the nighttime simulation.

Method	Scene Type	Illumination Conditions	RGB Only	Nighttime Simulation
NeRFFactor [85]	Object	Multi	Yes	
TensoIR [28]	Object	Single	Yes	
InvRender [87]	Object	Single	Yes	
NeRF-OSR [60]	Front-Facing	Multi	Yes	
FEGR [73]	Large Scene	Single/Multi	LiDAR	
LightSim [56]	Large Scene	Single/Multi	LiDAR	
UrbanIR (Ours)	Large Scene	Single	Yes	✓

Table 1. Comparison of various recent relightable NeRF methods. UrbanIR is among the first to offer single-illumination and RGB-only relightable NeRF capabilities suitable for large-scale scenes.

Relightable Neural Fields: Relightable neural radiance field methods [8, 9, 24, 54, 74, 77, 81, 85] aim to factor the neural field into multiple intrinsic components and leverage neural shading equations for illumination and material modeling. These methods admit realistic and controllable rendering of scenes with varying lighting conditions and materials. However, most relightable NeRF methods focus on objects with surrounding views or small bounded indoor environments. Important exceptions are: NeRF-OSR [60], which assumes access to multiple lighting sources for decomposition, and NeRF meets explicit geometry [74], which either uses multiple lighting or exploits depth sensing, such as LiDAR. In contrast, our proposed approach only requires a single video captured under the same, unknown illumination, making it more applicable to a broader range of scenes.

Differentiable rendering techniques make inverse graph-

ics tasks more flexible and convenient [12, 44, 55]. Most render meshes are suitable for object-level rendering, and so are challenging to apply to large urban scenes. In contrast, we leverage neural radiance fields (NeRF) [51].

Shadow modeling using images is challenging. Methods trained to cast shadows from images [71, 84] are tailored for particular objects (pedestrians, cars, etc). Learned methods can detect and remove shadows from 2D images [21, 22, 70]. But inverse graphics require modeling the full 3D geometry, intrinsic scene properties, and ensuring temporal consistency. Model-based optimization methods can infer shadows but rely on accurate scene geometry [33, 67, 75]. Using visibility fields to model shadows results in difficulty providing consistent shadows in relation to the underlying geometry [60, 66, 76, 87]. In contrast, our method combines the strengths of learning-based monocular shadow prediction and removal and model-based inverse graphics.

References

[1] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-time rendering*. AK Peters/crc Press, 2019. 4

[2] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2014. 7

[3] H.G. Barrow and Joan M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, 1978. 7

[4] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. 2014. 7

[5] Anand Bhattad and D. A. Forsyth. Stylitgan: Prompting stylegan to produce new illumination conditions, 2023. 7

[6] Anand Bhattad, Daniel McKee, Derek Hoiem, and DA Forsyth. Stylegan knows normal, depth, albedo, and more. *arXiv preprint arXiv:2306.00987*, 2023. 7

[7] James F Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, 1977. 2, 4

[8] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, 2021. 8

[9] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 8

[10] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 5, 6, 8

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 5, 6

[12] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in neural information processing systems*, 32, 2019. 9

[13] Zhihao Chen, Lei Zhu, Liang Wan, Song Wang, Wei Feng, and Pheng-Ann Heng. A multi-task mean teacher for semi-supervised shadow detection. In *CVPR*, 2020. 2, 4

[14] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5, 6

[15] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 4

[16] Dawson-Haggerty et al. trimesh. 4

[17] Yue Dong, Guojun Chen, Pieter Peers, Jiawan Zhang, and Xin Tong. Appearance-from-motion: Recovering spatially varying surface reflectance under unknown lighting. *ACM Transactions on Graphics (TOG)*, 33(6):1–12, 2014. 7

[18] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, 2021. 3, 4

[19] David Forsyth and Jason J Rock. Intrinsic image decomposition using paradigms. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7624–7637, 2021. 7

[20] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009. 7

[21] Lanqing Guo, Chong Wang, Wenhan Yang, Siyu Huang, Yufei Wang, Hanspeter Pfister, and Bihan Wen. Shadowdiffusion: When degradation prior meets diffusion model for shadow removal. *arXiv preprint arXiv:2212.04711*, 2022. 9

[22] Lanqing Guo, Siyu Huang, Ding Liu, Hao Cheng, and Bihan Wen. Shadowformer: Global context helps image shadow removal. *AAAI*, 2023. 2, 4, 9

[23] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3, 6

[24] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *arXiv:2206.03380*, 2022. 8

[25] Daniel Hauagge, Scott Wehrwein, Kavita Bala, and Noah Snavely. Photometric ambient occlusion. In *CVPR*, 2013. 7

[26] Berthold KP Horn. Determining lightness from an image. *Computer graphics and image processing*, 1974. 7

[27] Berthold KP Horn. Obtaining shape from shading information. *The psychology of computer vision*, 1975. 7

[28] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. *CVPR*, 2023. 4, 8

[29] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 1984. 2

[30] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *CVPR*, 2022. 3

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 4

[32] Pierre-Yves Laffont, Adrien Bousseau, and George Drettakis. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE transactions on visualization and computer graphics*, 2012. 7

[33] Samuli Laine, Timo Aila, Ulf Assarsson, Jaakko Lehtinen, and Tomas Akenine-Möller. Soft shadow volumes for ray tracing. In *ACM SIGGRAPH 2005 Papers*, pages 1156–1165. 2005. 9

[34] Jean-François Lalonde and Iain Matthews. Lighting estimation in outdoor image collections. In *International Conference on 3D Vision (3DV)*. IEEE, 2014. 1

[35] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 1971. 7

[36] Hendrik PA Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *TOG*, 2003. 7

[37] Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Extreme weather synthesis in neural radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3227–3238, 2023. 4

[38] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: SVBRDF acquisition with a single mobile phone image. In *ECCV*, pages 72–87, 2018. 7

[39] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

[40] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR*, 2020.

[41] Zhengqin Li, Jia Shi, Sai Bi, Rui Zhu, Kalyan Sunkavalli, Miloš Hašan, Zexiang Xu, Ravi Ramamoorthi, and Manmohan Chandraker. Physically-based editing of indoor scene lighting from a single image. *ECCV*, 2022. 7

[42] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *in arXiv*, 2021. 3, 4, 5

[43] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W Jacobs. Shape and material capture at home. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6123–6133, 2021. 7

[44] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 9

[45] Stephen Lombardi and Ko Nishino. Reflectance and illumination recovery in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):129–141, 2015. 7

[46] Stephen Lombardi and Ko Nishino. Radiometric scene decomposition: Scene reflectance, illumination, and geometry from rgb-d images. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 305–313. IEEE, 2016.

[47] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):65, 2019. 7

[48] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 1987. 5

[49] Wei-Chiu Ma, Hang Chu, Bolei Zhou, Raquel Urtasun, and Antonio Torralba. Single image intrinsic decomposition without a single intrinsic image. In *ECCV*, 2018. 7

[50] Stephen Robert Marschner. *Inverse rendering for computer graphics*. Cornell University, 1998. 7

[51] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 9

[52] Thomas Müller. tiny-cuda-nn, 2021. 4

[53] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 2022. 1, 4

[54] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. *arXiv:2111.12503 [cs]*, 2021. 8

[55] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, 2022. 7, 9

[56] Ava Pun, Gary Sun, Jingkang Wang, Yun Chen, Ze Yang, Sivabalan Manivasagam, Wei-Chiu Ma, and Raquel Urtasun. Neural lighting simulation for urban scenes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 4, 8

[57] Yi-Ling Qiao, Alexander Gao, Yiran Xu, Yue Feng, Jia-Bin Huang, and Ming C Lin. Dynamic mesh-aware radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 385–396, 2023. 4

[58] Chen Quei-An. ngp_pl: a pytorch-lightning implementation of instant-ngp, 2022. 1

[59] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 5, 6

[60] Viktor Rudnev, Mohamed Elgharib, William Smith, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. Nerf for outdoor scene relighting. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 6, 8, 9

[61] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination from shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003. 7

[62] Yoichi Sato, Mark D Wheeler, and Katsushi Ikeuchi. Object shape and reflectance modeling from observation. In *SIGGRAPH*, 1997. 7

[63] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 6

[64] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 6

[65] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 7

[66] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021. 9

[67] Jon Story. Hybrid ray traced shadows. In *Game Developer Conference*, 2015. 9

[68] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 7

[69] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022. 3

[70] Jin Wan, Hui Yin, Zhenyao Wu, Xinyi Wu, Yanting Liu, and Song Wang. Style-guided shadow removal. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*, pages 361–378. Springer, 2022. 9

[71] Yifan Wang, Andrew Liu, Richard Tucker, Jiajun Wu, Brian L Curless, Steven M Seitz, and Noah Snavely. Repopulating street scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5110–5119, 2021. 9

[72] Zian Wang, Wenzheng Chen, David Acuna, Jan Kautz, and Sanja Fidler. Neural light field estimation for street scenes with differentiable virtual object insertion. In *ECCV*, 2022. 5

[73] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *CVPR*, 2023. 1, 2, 4, 5, 8

[74] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representation for inverse rendering of urban scenes. *arXiv*, 2023. 8

[75] Tai-Pang Wu, Chi-Keung Tang, Michael S Brown, and Heung-Yeung Shum. Natural shadow matting. *ACM Transactions on Graphics (TOG)*, 26(2):8–es, 2007. 9

[76] Wenqi Yang, Guanying Chen, Chaofeng Chen, Zhenfang Chen, and Kwan-Yee K Wong. S ̂3-nerf: Neural reflectance field from shading and shadow under a single viewpoint. *arXiv preprint arXiv:2210.08936*, 2022. 9

[77] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019. 7, 8

[78] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999. 7

[79] Ye Yu, Abhimitra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and William A. P. Smith. Self-supervised outdoor scene relighting. In *ECCV*, 2020. 6

[80] Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenman, and Jean-François Lalonde. All-weather deep outdoor lighting estimation. In *CVPR*, 2019. 1

[81] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *CVPR*, 2021. 8

[82] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *CVPR*, 2022. 7

[83] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE TPAMI*, 1999. 7

[84] Shuyang Zhang, Runze Liang, and Miao Wang. Shadowgan: Shadow synthesis for virtual objects with conditional adversarial networks. *Computational Visual Media*, 2019. 9

[85] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM TOG*, 2021. 2, 4, 8

[86] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv*, 2020. 7

[87] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 2, 4, 8, 9

[88] Tinghui Zhou, Philipp Krahenbuhl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *ICCV*, 2015. 7