The Sky's the Limit: Relightable Outdoor Scenes via a Sky-pixel Constrained Illumination Prior and Outside-In Visibility

James A. D. Gardner[1], Evgenii Kashin[1], Bernhard Egger[2], and William A. P. Smith[1]

[1] Department of Computer Science, The University of York, York, YO10 5DD, UK
{james.gardner,evgenii.kashin,william.smith}@york.ac.uk
[2] Cognitive Computer Vision Lab, Friedrich-Alexander-Universität, Erlangen-Nürnberg, Erlangen, Germany
bernhard.egger@fau.de

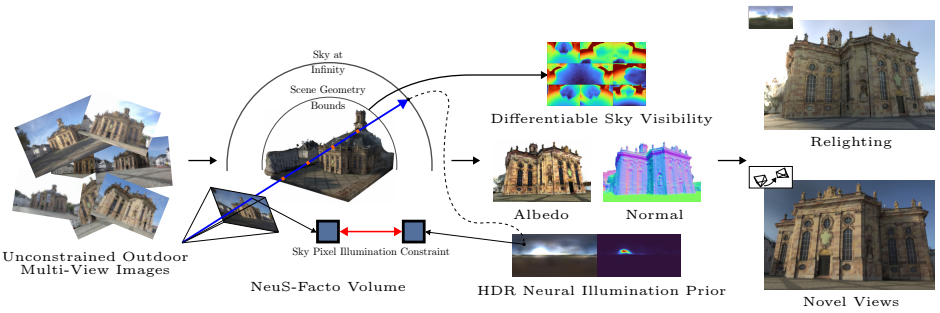

Fig. 1: From in-the-wild, outdoor image collections, we predict scene geometry, albedo, distant environment illumination, and sky visibility. Sky visibility and illumination are both modelled via spherical neural fields whereby we directly constrain illumination via sky pixel observations. Our outside-in differentiable visibility enables estimation of cast shadows and avoids shadow baking into albedo.

Abstract. Inverse rendering of outdoor scenes from unconstrained image collections is a challenging task, particularly illumination/albedo ambiguities and occlusion of the illumination environment (shadowing) caused by geometry. However, there are many cues in an image that can aid in the disentanglement of geometry, albedo and shadows. Whilst sky is frequently masked out in state-of-the-art methods, we exploit the fact that any sky pixel provides a direct observation of distant lighting in the corresponding direction and, via a neural illumination prior, a statistical cue to derive the remaining illumination environment. The incorporation of our illumination prior is enabled by a novel 'outside-in' method for computing differentiable sky visibility based on a neural directional distance function. This is highly efficient and can be trained in parallel with the neural scene representation, allowing gradients from appearance loss to flow from shadows to influence the estimation of illumination and geometry. Our method estimates high-quality albedo, geometry, illumination and sky visibility, achieving state-of-the-art results on the NeRF-OSR relighting benchmark. Our code and models can be found at https://github.com/JADGardner/neusky.

1 Introduction

Inverse rendering of outdoor scenes has diverse downstream applications such as scene relighting, augmented reality, game asset generation, and environment capture for films and virtual production. However, accurately estimating the underlying scene model that produced an image is an inherently ambiguous task due to its ill-posed nature [2]. To address this, many works use some combination of handcrafted [7, 17] or learned priors [4, 6, 19, 47], inductive biases in model architectures [13], or multi-stage training pipelines [35, 36, 50]. This process is made even more difficult when considering in-the-wild image collections from the internet that contain transient objects, image filters, unknown camera parameters and changes in illumination.

Outdoor scenes present particular challenges. Natural illumination from the sky is complex and exhibits an enormous dynamic range. This causes strong cast shadows when the brightest parts of the sky are occluded. These occlusions are non-local and discontinuous making them hard to incorporate within a differentiable renderer. Outdoor scene geometry can also exhibit arbitrary ranges of scale. On the other hand, sky illumination dominates secondary bounce lighting, meaning it is reasonable to assume a spatially non-varying, distant illumination environment. In addition, natural illumination contains statistical regularities [14] that make it easier to model. For example, luminance generally increases with elevation (the 'lighting-from-above' prior), the sun can only be in one position and the range of possible colours from sun and sky light is limited.

In this paper, we tackle the outdoor scene inverse rendering problem by fitting a neural scene representation to a multi-view, varying-illumination photo collection. We name our method *NeuSky*, and make four key contributions relative to prior work. First, we make a key observation: Any pixel in an image that observes the sky provides a direct constraint on the illumination environment in that direction. Second, we combine this insight with an HDR neural field natural illumination model [16] learnt from natural environments, constraining this model to outpaint plausible illuminations given the direct observations of illumination seen from the camera. Thirdly, we propose *outside-in visibility*, a novel, differentiable, neural approximation to sky visibility, computed with a single forward pass through a directional distance function network. Finally, we deploy this visibility representation to enable end-to-end training, removing the need for phased training. Crucially, this means that shadows can influence illumination and geometry estimation by appearance losses backpropagating through the visibility network, enabling geometry estimation for non-observed scene regions and also avoiding shadow baking into albedo.

2 Related Work

Relightable Neural Scenes The core NeRF [26] approach has been improved in several key ways since its publication. Nerfstudio, [37] a platform for researching in Neural Fields, introduced NeRFacto taking advantage of many of these

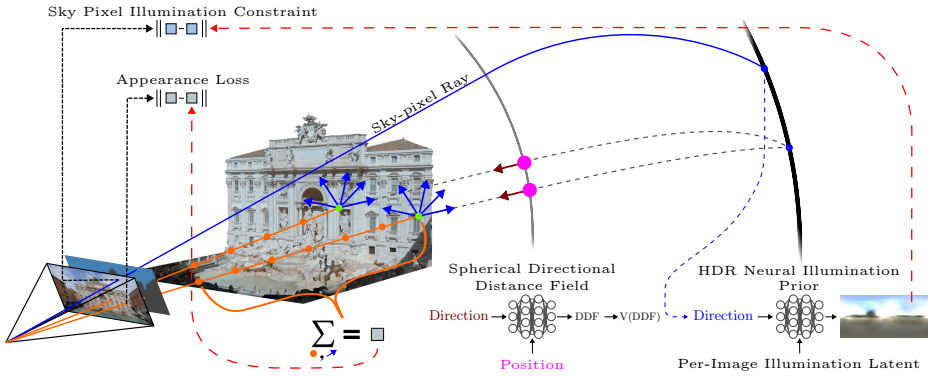

Fig. 2: We surround our NeuS-Facto [48] volume with two spherical neural fields at radius 1 and radius ∞ modelling sky visibility and distant illumination respectively. Blue arrows correspond to rays sampling distant illumination. Pink circles and Maroon arrows are position and direction samples of our sky visibility network. In a given direction, visibility changes with position but distant illumination does not. For speed we only sample sky visibility on the surface of our scene, Green circles, and distribute this visibility to all samples, Orange circles, along a ray.

developments. It leverages the same proposal sampling and scene contraction as Mip-NeRF 360 [3] alongside the hash-grid representation from Instant-NGP [27] to reduce network sizes and vastly speed up training. Implicit surface representations were introduced in NeuS [39] and VolSDF [45], which used a neural Signed Distance Function (SDF) with NeRF volume rendering. NeuS-Facto, introduced in SDFStudio [48], combined the NeRFacto improvements with NeuS. This model, which is similar to that used by the current state-of-the-art in neural surface reconstruction of large scenes [21], is the underlying model that we use.

In parallel with these developments, several attempts have been made to use neural scene representations for decomposition into its intrinsic properties. NeRF-OSR [30] predicts albedo and density. For distant illumination, they predict per image Spherical Harmonic (SH) lighting coefficients and model shadows via a shadow network conditioned on those SH coefficients. Whilst now providing a parametric model of illumination they are limited by the quality of normals obtained from NeRF density (we use a NeuS derivative with high-quality geometry), shadows that are not related to the scene geometry (our shadow network is directly tied to scene geometry) and the low frequency of SH (we employ a neural field for illumination capable of capturing higher order lighting effects). Methods such as PhySG [49] and NeRF-V [35] allow relighting but require known illumination. NeRFactor [50] additionally optimises visibility and illumination together allowing shadows but with a low-resolution environment map and no illumination prior. Similar to our work, FEGR [41] also uses a neural field representation for HDR illumination, however, they do not include a prior over illuminations. Their rasterisation process to model visibility is also a non-differentiable function, meaning cues from shading and shadows will not

inform illumination or geometry estimations. SOL-NeRF [36] similarly convert their SDF representation to a mesh for ray-tracing but instead use a combination of Spherical Gaussians (SG), with a sunlight colour prior based on sun elevation, and SH to model illumination. Also similar to our work, NeuLighting [20] uses a prior over illuminations and visibility MLP but their framework is trained in a cascaded manner, so visibility can not influence lighting and geometry estimations compared to our method, furthermore their method considers shadows only from the sun.

Directional Distance Fields SDFs measure the distance to the nearest surface at a given point, signed to indicate outside/inside. In contrast, Directional Distance Functions (DDFs) measure the distance to the nearest surface *in a given direction*, making them 5D as opposed to 3D functions for SDFs. Interest in DDFs has primarily been as a geometry representation that allows faster rendering (no sphere tracing is required). Neural DDFs were primarily introduced in [52] which developed the Signed Directional Distance Functions (SSDF) as a model of continuous distance view synthesis and derived many important properties of SDDFs. This was later extended by [1], which enabled the modelling of internal structures via dropping the sign and extending the representation via probabilistic modelling. Subsequent works enable to model of shapes with no explicit boundary surface [38], refine the multi-view consistency of DDFs [22] and employ SDDFs to improve optimisation of multi-view shape reconstruction [51]. Our usage of a DDF is most similar to that of FiRE [46] which also combines an SDF scene representation with a DDF sampled only on the unit sphere. However, unlike FiRE, whose goal was fast rendering, we show how to use a spherical DDF for fast, differentiable sky visibility. A more in-depth explanation of DDFs is found in Section 3.2.

Neural Illumination and Visibility Boss et al. [5] proposed neural pre-integrated lighting (PIL), a spherical neural field conditioned on a roughness parameter to model an illumination environment convolved with a BRDF. This enabled fast rendering but at the expense of being unable to model occlusions of the illumination environment. RENI [15], proposed by Gardner et al., is a vertical-axis rotation-equivariant conditional spherical neural field, trained on thousands of HDR outdoor environment maps to learn a prior for natural illumination. The low-dimensional but expressive latent space is useful for constraining inverse rendering problems. This was subsequently extended in RENI++ [16] with the addition of scale-invariant training and a transformer-based architecture. Several other recent methods aim to predict illumination from small image crops [12,34], as a 5D light field network [44], from a text description [9] or using diffusion models with differentiable path tracing [25]. Rhodin et al. [29] approximate scene geometry with Gaussian blobs for differentiable visibility. Lyu et al. [24] similarly use spheres for geometry and model illumination with spherical harmonics for approximate differentiable shadows. Worchel and Alexa [43] use a differentiable mesh renderer for classical shadow mapping [42].

3 Method

Our method takes as input a dataset of N images. From these images we compute poses with COLMAP [31, 32] and semantic segmentation maps with ViT-Adapter [10] according to the Cityscapes [11] convention. The preprocessed dataset comprises $\mathcal{D} = \{(\mathcal{I}_i, \mathbf{E}_i, \mathbf{K}_i, \mathcal{S}_i)\}_{i=1}^{N}$, where $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ is an image, $\mathcal{S} \in \mathbb{Z}^{H\times W}$ is the segmentation map and $\mathbf{E} = [\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{3\times 4}$ and $\mathbf{K} \in \mathbb{R}^{3\times 3}$ are the camera extrinsics and intrinsics respectively. To align the vertical axis of our scene with gravity, we robustly fit a plane to the camera positions and rotate to align with the x-y plane.

Scene Representation We model scene geometry as a neural SDF, such that at any point $\mathbf{x} \in \mathbb{R}^3$, the signed distance is given by $f_{\text{SDF}}(\mathbf{x}) \in \mathbb{R}$. We assume that the scene is Lambertian, with diffuse albedo modelled by the neural field $\mathbf{a}(\mathbf{x}) \in [0, 1]^3$. We also assume that illumination is a distant environment that depends only on direction $\mathbf{d} \in S^2$, with HDR RGB incident radiance given by $L_i(\mathbf{d}) \in \mathbb{R}^3_{\geq 0}$.

Rendering We follow NeuS [40] and derive a volume density, $\sigma(\mathbf{x})$, from the SDF value. This allows volume rendering of the SDF in the same fashion as in NeRF. For a ray $\mathbf{r}$ with origin $\mathbf{o}$ and direction $\mathbf{v}$, the time-discrete volume rendered RGB colour is given by:

$$\mathbf{c}(\mathbf{r}) = \sum_{j=1}^{S} w_j \mathbf{a}(\mathbf{x}_j) \sum_{k=1}^{D} L_i(\mathbf{d}_k) V(\mathbf{x}_E, \mathbf{d}_k) \max(0, \mathbf{n}(\mathbf{x}_j) \cdot \mathbf{d}_k), \tag{1}$$

where the first summation is over the S samples along the ray, while the second is over the D lighting direction samples. The lighting direction samples are distributed approximately uniformly over the sphere by using an 8-subdivided icosahedron giving $D = 642$. w_j is the volume rendering blending weight for the jth sample point which depends on $t_{1\ldots j}$ and $\sigma(\mathbf{x}_{1\ldots j})$, with $\mathbf{x}_j = \mathbf{o} + t_j\mathbf{v}$. $V(\mathbf{x}, \mathbf{d}) \in \{0, 1\}$ is the sky visibility in direction $\mathbf{d}$ at position $\mathbf{x}$ with $\mathbf{x}_E$ being the position at the expected termination depth of the ray $\mathbf{x}_j$. $\mathbf{n}(\mathbf{x}) = \nabla f_{\text{SDF}}(\mathbf{x})/\|\nabla f_{\text{SDF}}(\mathbf{x})\|$ is the surface normal at $\mathbf{x}$, derived from the gradient of the SDF.

We define our appearance loss for a batch of rays $\mathcal{R}$ as:

$$\mathcal{L}_{\text{app}} = \sum_{\mathbf{r}\in\mathcal{R}} \ell(\mathbf{c}_{\text{gt}}(\mathbf{r}), \text{sRGB}(\mathbf{c}(\mathbf{r}))), \tag{2}$$

where $\mathbf{c}_{\text{gt}}(\mathbf{r})$ is the ground truth colour for ray $\mathbf{r}$, sRGB$(\cdot)$ tonemaps the linear image provided by our model and ℓ computes the sum of L1 and cosine errors (to match both absolute RGB values and hue). To avoid overfitting we apply a random rotation $R \sim \mathcal{U}(SO(3))$ to jitter the direction vectors $\mathbf{d}_k$ in every batch.

Neural Illumination Model To restrict L_i to the space of plausible natural illumination environments, we use a neural illumination prior, RENI++ [16]. This is a conditional neural field, $f_{L_i} : S^2 \times \mathbb{R}^{3\times K} \to \mathbb{R}^3$ that outputs log

HDR RGB colours in the given input direction, conditioned on a normally distributed 3D latent code $\mathbf{Z} \in \mathbb{R}^{3\times K}, \text{vec}(\mathbf{Z}) \sim \mathcal{N}(\mathbf{0}_{3K}, \mathbf{I}_{3K})$. The latent space of RENI++ provides a low dimensional characterisation of natural, outdoor illumination environments since it was trained on several thousand real-world outdoor environment maps. This provides useful global constraints on the estimated illumination, which is only partially observable in any one image. In addition, the normally-distributed latent space provides a prior while the latent code is vertical-axis rotation-equivariant (rotating $\mathbf{Z}$ about the vertical axis corresponds to similarly rotating the environment). This vertical axis corresponds to gravity and we therefore align the vertical axis of our scene with gravity as described above. We optimise a RENI++ latent code, $\mathbf{Z}_i$, and absolute scale, γ_i, for each image i in the training set and replace $L_i(\mathbf{d}_k)$ with $\gamma_i \exp(f_{L_i}(\mathbf{d}_k, \mathbf{Z}_i))$ in (1). To ensure the estimated illumination is plausible, we include a prior loss: $\mathcal{L}_{\text{prior}} = \|\mathbf{Z}\|_2^2$ for all latent codes. In Section 3.1 we describe how the illumination environment in an image can be additionally constrained via sky pixel observations.

Reducing Visibility Tests Visibility of the illumination environment from a scene point is required in our rendering equation (1) and is essential for recreating cast shadows and ambient occlusion effects. However, computing sky visibility from a neural SDF is computationally expensive. It requires sphere tracing from the query point in the light direction until the ray hits another part of the surface or leaves the scene bounds. To render a single pixel, this must be performed D times for each of the S sample points. We therefore propose two methods to drastically reduce the number of visibility samples. First, since we are only concerned with visibility on the surface of the scene, we define $\mathbf{x}_E = \mathbf{o} + t_E\mathbf{v}$, where t_E is the current expected termination depth of the ray, and evaluate visibility only at $\mathbf{x}_E$. This means we only need D visibility tests per pixel since we reuse the computed visibilities for all sample points along the ray. Second, any light direction in the lower hemisphere, i.e. where $(\mathbf{d}_j)_z < 0$, will strike either the scene or the ground. For these directions we set $V(\cdot) = 1$, i.e. visible. The rationale for this is that the RENI++ illumination environment will learn to capture the colour of the ground or lower hemisphere of the scene, averaged over all spatial positions. This provides an approximation to secondary illumination from the ground. We found this to perform considerably better than setting these directions as non-visible. In spite of these two speedups, the remaining $D/2$ visibility tests still prove too expensive if performed via sphere tracing of the SDF. For this reason, in Section 3.2 we propose a fast, softened approximation for visibility.

3.1 Sky Pixel Constrained Illumination Prior

Pixels labelled in the semantic segmentation maps with the 'sky' class (hereafter referred to as *sky pixels*) provide a direct observation of the distant illumination environment in the direction given by the ray for that pixel. To the best of our knowledge, this constraint has never been used to aid illumination estimation

in inverse rendering methods. Since our illumination model, RENI++ [16], captures the space of plausible natural illuminations, even observing only a portion of the sky provides a strong statistical cue. For example, if a bright region corresponding to the sun is observed, then RENI++ cannot create another sun in an unobserved part of the environment. Alternatively, if all observed sky is white, it is likely to be an overcast day and RENI++ will predict an ambient environment without a discernible sun. Using sky pixel constraints alone can be viewed as statistical outpainting of the whole environment from the portion observed in an image. In practice, we incorporate this within our inverse rendering framework such that the appearance loss of non-sky pixels also provides a rich, indirect constraint on the illumination.

The sky segmentation also provides an additional constraint that is similar to the widely used mask loss. Since we know that sky ray pixels miss the scene, we penalise our neural scene representation from placing any density along the ray, providing geometric supervision. Together, these form our sky loss:

$$\mathcal{L}_{\text{sky}} = \sum_{\mathbf{r} \in \mathcal{R} \cap \mathcal{S}_{\text{sky}}} \varepsilon(\mathbf{c}_{\text{gt}}(\mathbf{r}), \mathbf{c}_{\text{sky}}(\mathbf{r})) - \log(1 - \sum_j w_j), \tag{3}$$

where $\mathcal{S}_{\text{sky}}$ is the set of sky pixels. The first term is the error between the observed sky pixel colour and predicted, $\mathbf{c}_{\text{sky}}(\mathbf{r}) = \text{sRGB}(\gamma \exp(f_{L_i}(\mathbf{r}, \mathbf{Z})))$, and the second term is the binary cross entropy loss on the accumulated density in sky pixels.

3.2 Outside-in Sky Visibility

Shadows offer a wealth of information about geometry, both within and beyond the view frustum. For instance, if the sun is predicted to be behind the camera and a prominent cast shadow appears on the floor, we can infer there is geometry behind the camera and the likely sun direction. However, to fully leverage this information it is necessary to have a differentiable model of visibility.

To address this, we draw inspiration from works, NeRFactor [50] and NeRV [35] and learn a neural model of visibility. However, to make training tractable, [50] learn their visibility representation in a second training phase with geometry pretrained and frozen and [35] require known illumination. Initial attempts to model visibility using the same parameterisation as [35] were unable to fit in our less constrained and end-to-end task. We desire a model of visibility that is consistent with the geometry of our scene, fast to sample from and differentiable, enabling gradients from visibility to inform illumination, albedo and geometry estimation. However, this model must be constrained enough that training end-to-end with our scene representation is tractable. To achieve this, we propose *outside-in visibility* in which visibility is represented implicitly via a Spherical Directional Distance Field (SDDF) defined on the radius 1 sphere that bounds our scene and is tied to our SDF scene representation via consistency losses. Our geometric volume is represented with the Mip-NeRF 360 [3] scene contraction. This means that parallel rays converge to a point on the radius 2 sphere

(representing infinity). Hence, our visibility model resides on the radius 1 sphere where position-dependent visibility can be reasoned about, while our distant illumination model is defined on the radius 2 sphere (see Figure 3).

Spherical Directional Distance Function Consider a point $\mathbf{s} \in S^2$ lying on a bounding sphere of radius 1. The Spherical Directional Distance Function (DDF), $f_{\text{DDF}} : S^2 \times S^2 \rightarrow \mathbb{R}$, returns the (positive) distance from $\mathbf{s}$ for any inward-pointing direction $\mathbf{d}$ to the first intersection with the surface. In other words, the spherical DDF stores an inward looking depth map of the scene from any viewpoint on the radius r sphere. The DDF is related to the SDF: $f_{\text{SDF}}(\mathbf{s} + f_{\text{DDF}}(\mathbf{s}, \mathbf{d})\mathbf{d}) = 0$, such that moving the distance given by the DDF must arrive at the surface where the SDF is zero. However, there may be multiple such points and the DDF must return the minimum, giving us another constraint: $f_{\text{DDF}}(\mathbf{s}, \mathbf{d}) = \min\{t | f_{\text{SDF}}(\mathbf{s} + t\mathbf{d}) = 0\}$.

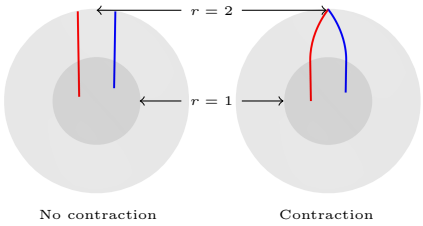

Fig. 3: We model our illumination and illumination visibility via two spherical neural fields at radius ∞ and 1 respectively. However our world space is contracted as per Mip-NeRF-360 [3], such that any point at infinity is placed on the sphere of radius 2. Since we model distant illumination, the sampled colour only depends on direction, and two samples at different locations but in the same direction will sample RENI++ [16] at the same point. However, visibility of distant illumination *is* dependent on location and the intersection of the ray on the sphere of radius 1 is used to sample our visibility network.

The DDF is required to learn a very complex function: essentially an inward-facing depth map of the scene from any position on the sphere. We found that this function is easier to learn if we define a consistent coordinate frame to parameterise directions for any given point on the sphere. We normalise the inward-facing directions from world coordinates to a local coordinate system such that the y-axis aligns with $\mathbf{s}$ (the sample position on the DDF), the x-axis is orthogonal to y and to our world-up, and the z-axis is orthogonal to y and x. See Figure 7 for a visualisation.

Sky Visibility via Directional Distance Fields Our key insight is to show how to use the inward looking DDF as a representation for computing outward sky visibility (see Figure 4). Consider a point $\mathbf{x} \in \mathbb{R}^3$ lying on the surface (and inside the bounding sphere, such that $\|\mathbf{x}\| \leq 1$). We can use the DDF to check whether $\mathbf{x}$ can see the sky or is occluded in a direction $\mathbf{d}$. First we compute the point $\mathbf{s}$ as the solution to $\mathbf{s} = \mathbf{x} + t\mathbf{d}$, s.t. $\|\mathbf{s}\| = 1$ and $t \geq 0$, i.e. the point on the radius r sphere that is intersected by the ray in direction $\mathbf{d}$ from $\mathbf{x}$. Next, we evaluate the DDF at $\mathbf{s}$ in direction $-\mathbf{d}$ (i.e. outside-in): $f_{\text{DDF}}(\mathbf{s}, -\mathbf{d})$. If $\mathbf{x}$ is not occluded then the DDF value should be similar to the actual distance between $\mathbf{s}$ and $\mathbf{x}$: $f_{\text{DDF}}(\mathbf{s}, -\mathbf{d}) \approx \|\mathbf{s} - \mathbf{x}\|$. However, if $\mathbf{s}$ is occluded then the DDF will return a distance significantly less than the actual distance: $f_{\text{DDF}}(\mathbf{s}, -\mathbf{d}) < \|\mathbf{s} - \mathbf{x}\|$. Binary visibility can be computed by testing whether this difference is below

a threshold ϵ: $V = (\|\mathbf{s} - \mathbf{x}\| - f_{\text{DDF}}(\mathbf{s}, -\mathbf{d}) < \epsilon)$. Note that this is equivalent to classical shadow mapping [42] with the exception that we rely on a DDF forward pass as opposed to (non-differentiable) rasterisation of a mesh from the light source perspective.

However, binary visibility is discontinuous and so not suitable for propagating loss gradients through visibility and back into geometry. For this reason, we replace the discrete threshold with a softened approximation (see Figure 5):

$$V(\mathbf{x}, \mathbf{d}) = 1 - \kappa\left(\eta(\|\mathbf{s} - \mathbf{x}\| - f_{\text{DDF}}(\mathbf{s}, -\mathbf{d}) - \epsilon)\right), \tag{4}$$

where κ is the sigmoid function. The threshold ϵ controls the tolerance on what is considered a shadow. We make this learnable and initialise it with a large value (equal to the scene radius). When ϵ is large, no parts of the scene will be considered occluded. As training converges, ϵ can be reduced to gradually introduce more illumination occlusions. The parameter s controls the sharpness of the transition between occluded and unoccluded.

Supervising the DDF The DDF indirectly determines visibility which in turn determines appearance via the rendering equation in (1). This means that the DDF is partially supervised by the appearance loss. However, we also require that the DDF's representation of scene geometry is consistent with the SDF geometry. We enforce this consistency through four losses. First, $\mathcal{L}_{\text{ddf\_depth}}$, enforces that the depth predicted by the DDF should match that of the scene parameterised by the SDF. Second, $\mathcal{L}_{\text{ddf\_levelset}}$, ensures that travelling the distance predicted by the DDF should arrive at the SDF zero level set. Third, we encourage multiview consistency in the DDF via a multiview consistency loss $\mathcal{L}_{\text{ddf\_multiview}}$. Finally, with, $\mathcal{L}_{\text{ddf\_sky}}$, we further take advantage of our sky segmentation maps as an additional constraint on our DDF. Rays that intersect the sky have no occlusions between the camera origin and our DDF sphere. Our DDF should therefore predict at least the distance to the camera origin for those intersecting rays. Detailed descriptions of these losses can be found in the supplementary.

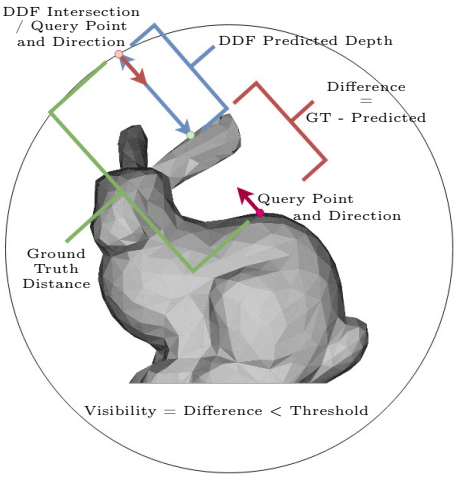

Fig. 4: Visibility of our neural illumination from a point in the scene is implicitly represented via our Directional Distance Field (DDF) which represents the depth to the surface of our scene from any point on the unit sphere. The DDF is a spherical neural field that surrounds our scene at radius 1. The DDF is fully differentiable allowing gradients obtained from shadowing to inform illumination and geometry.

3.3 Implementation

We implement our method in Nerfstudio [37], building on top of NeuS-Facto [48]. We convert our CityScapes [11] segmentation masks into classes for sky, ground plane, foreground and transient objects (vehicles, vegetation, people etc). We sample ray batches only from non-transient pixels. We use a hash grid with 16 levels, 2^{19} hash table size, 2 features per entry and a course and fine resolution of 16 and 2048 respectively. Our SDF and albedo networks are both 2-layer 256-neuron MLPs. We initialise our SDF as a sphere with radius=0.1. We use the pre-trained RENI++ [16] model with a latent dimension $K = 100$ and initialise latent codes as zeroes, corresponding to the mean environment provided by the RENI++ prior. We initialise the per-image illumination scale as $\gamma = 1$.

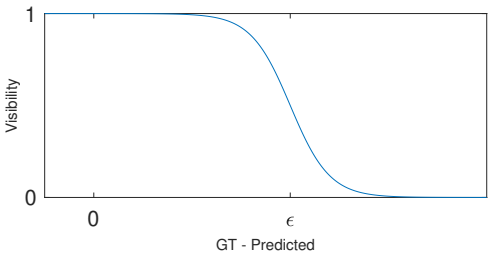

Fig. 5: Soft visibility function. We plot $\|\mathbf{s} - \mathbf{x}\| - f_{\text{DDF}}(\mathbf{s}, -\mathbf{d})$ on the x-axis versus $V(\mathbf{x}, \mathbf{d})$ on the y-axis. When ground truth distance is significantly smaller than the threshold ϵ, we assign a visibility of 1. When significantly larger, we infer an occlusion and a visibility of 0. In the vicinity of ϵ we smoothly transition from visible to non-visible with a steepness controlled by η.

Our visibility network is a FiLM-Conditioned [8] SIREN [33] with 5 layers and 256 neurons in both the FiLM Mapping Network and the main SIREN. We condition our network on positions on the sphere using the same dimensional hash grid as our SDF. We first map from position to a hashed latent, this latent is then provided to the FiLM mapping network to condition the model. As per Section 3.2 we normalise direction to a local coordinate frame and these directions are positionally encoded as per NeRF [26]. We use a sigmoid activation function scaled by the size of our scene bounds to ensure a depth prediction within the correct range. We generate samples for DDF supervision via our PyTorch re-implementation of fast von Mises-Fisher distribution sampling from Pinzón et al [28]. We used a concentration parameter of 20.0 for the distribution and sampled 8 positions and 128 directions per batch exclusively from the upper hemisphere.

We optimise our Proposal Samplers, SDF/Albedo Field and DDF using Adam [18] optimisers with a Cosine Decay [23] schedule and 500-step 'warm-up' phase. Our loss is the sum of $\mathcal{L}_{\text{app}}$, $\mathcal{L}_{\text{prior}}$, $\mathcal{L}_{\text{sky}}$, the four DDF supervision losses and a proposal sampler interlevel loss as per Mip-NeRF 360 [3]. Our initial learning rates are 1e−2, 1e−3 and 1e−4 respectively. Our RENI++ latent codes and the visibility threshold parameter use Adam [18] optimisers with an exponentially decaying learning rate which is initialised at 1e−2 and 1e−3 respectively.

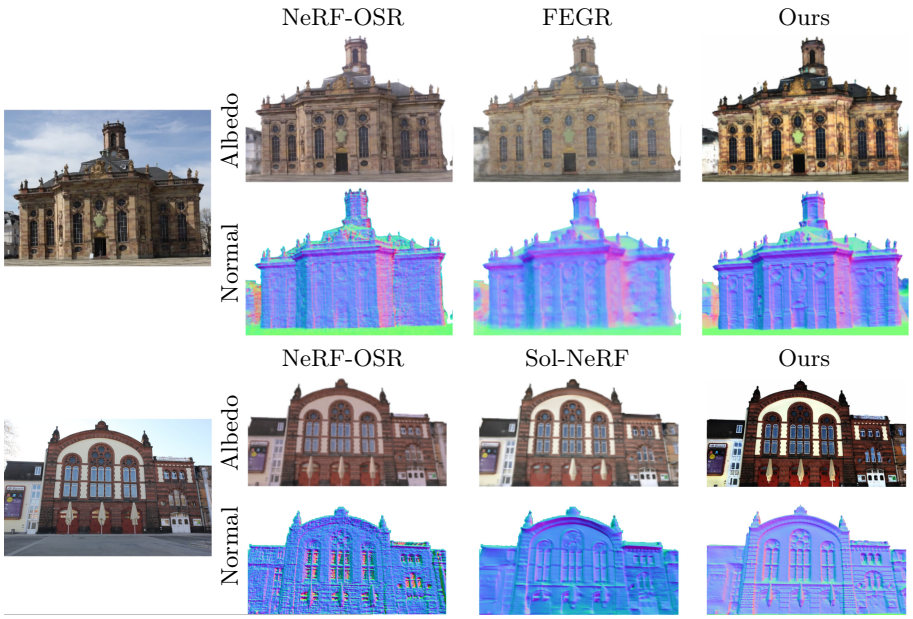

Fig. 6: Comparion of albedo and normals produced by NeRF-OSR [30], FEGR [41], SOL-NeRF [36] and our method. We produce much sharper albedo and normals than all prior works whilst training end-to-end.

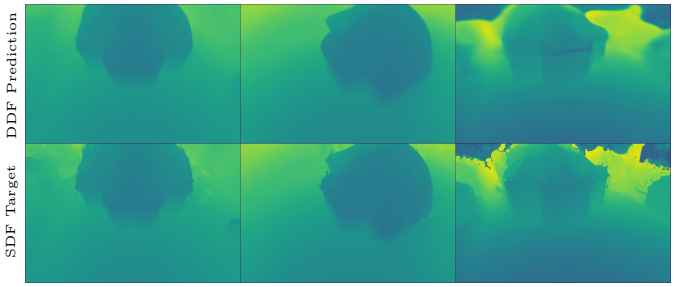

Fig. 7: Three views of the depth predicted by the Spherical DDF (top row) and its pseudo ground truth from the scene representation (bottom row) for *Site 1* in the NeRF-OSR [30] dataset (see Figure 1). Cameras are placed on the unit sphere looking towards the origin. The DDF is trained concurrently with the scene representation and can capture high-frequency details required for accurate shadows.

4 Evaluation

We begin by qualitatively evaluating the output of our system components. Figure 7 illustrates that our spherical DDF is able to produce detailed depth maps via a single forward pass from arbitrary viewpoints. The geometry of the building and ground plane are well reconstructed. In Figure 8 we visualise the output of the visibility network in two different ways. On the left we average visibility over all directions, giving a good approximation to ambient occlusion. On the right we compute visibility for a single direction, producing a sharp shadow.

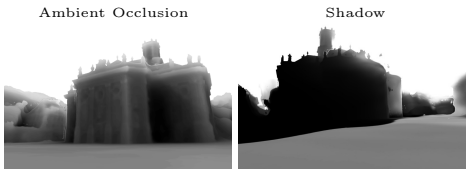

Fig. 8: Ambient occlusion and shadows from a point source computed from our soft visibility via the DDF.

Shadows Informing Geometry Due to our visibility model training concurrently with our scene representation, shadows can inform geometry outside of the view frustum. We can optionally apply stop gradients to visibility calculations to prevent this capability. We demonstrate the advantage of training end-to-end in Figure 13, which shows a rendering of *Site 1* looking behind the view frustums of all training cameras for that scene. To explain shadows seen during training geometry has been generated outside the view of any training camera. This is a key advantage of training our differentiable sky visibility network concurrently with our scene representation.

Without Visibility Network Figure 14 demonstrates the benefit of our sky-visibility network. When enabled our model is better able to disentangle shading from albedo, particularly in scenes in which many of the images captured are shaded, namely *Site 3* of the NeRF-OSR [30] dataset. Here, shadows on the ground, doors and building facade are removed from the albedo.

Relighting We evaluate NeuSky's relighting capabilities on the NeRF-OSR [30] relighting benchmark. The NeRF-OSR dataset consists of eight sites captured over multiple sessions each with differing illumination conditions along with Low Dynamic Range (LDR) ground truth environment maps. The benchmark test relighting three of these scenes. Our results can be found in Table 9. We achieve better relighting performance than both NeRF-OSR [30] and FEGR [41] and beat SOL-NeRF [36] on two of the three scenes. Qualitatively our method also produces models with significantly higher quality geometry and albedo than all three prior works, as shown in Figure 6. As shown in Figures 10 and 12, NeuSky is capable of disentangling illumination, albedo and shading and our sky visibility network and RENI++ combine to

	Site 1		Site 2		Site 3	
	PSNR ↑	MSE ↓	PSNR ↑	MSE ↓	PSNR ↑	MSE ↓
NeRF-OSR [30]	19.34	0.012	16.35	0.027	15.66	0.029
FEGR [41]	21.53	0.007	17.00	0.023	17.57	0.018
SOL-NeRF [36]	21.23	0.0084	**18.18**	**0.019**	17.58	0.028
NeuSky (Ours)	**22.50**	**0.005**	16.66	0.023	**18.31**	**0.016**

Fig. 9: Outdoor scene relighting results on the *NeRF-OSR* relighting benchmark.

produce sharp shadows. Further results are shown in Figure 1. In Figure 11 we demonstrate rendering scenes under novel illumination conditions.

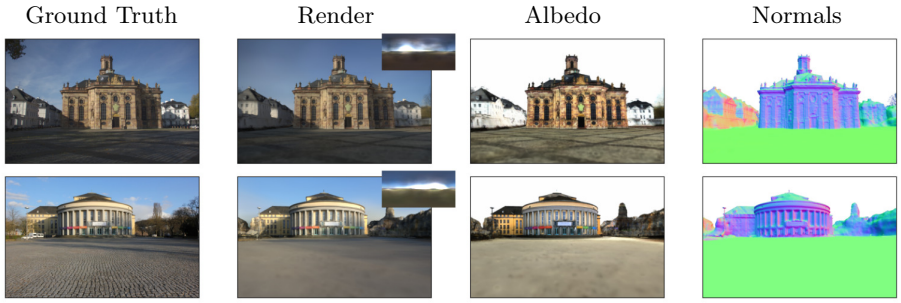

Fig. 10: A render from *Site-1* and *Site-2* in NeRF-OSR [30]. Environment maps sampled from the estimated illumination of RENI++ [16], albedo and normals are shown alongside the ground truth images. Our method accurately disentangles albedo, lighting and shadows whilst producing very high-quality geometry.

Fig. 11: Relighting under novel illuminations.

5 Conclusion

We have presented the first outdoor scene inverse rendering approach that incorporates a model of natural illumination, exploits direct sky pixel observations and can be trained end-to-end with a visibility model. This enables our model to reproduce accurate shadows, avoids shadow baking into albedo, allows shadows to constrain geometry and illumination and achieves superior geometry and albedo reconstruction on the NeRF-OSR dataset beating [30], [36] and [41] in the relighting benchmark. There are a number of limitations of our work, namely a high training GPU memory requirement when using large batches and at between 5-8 hours, our optimisation time is slow by modern neural field standards. The most obvious extension to our approach would be to use a more complex reflectance model and accompanying material parameters. For reflective surfaces, second bounce illumination becomes more significant. It is possible that a DDF could be used to speed up multibounce ray casting in this context.

Acknowledgments and Disclosure of Funding This project was supported by travel funds from the Bavarian Research Alliance (BayIntAn FAU -2023-29). James Gardner and Evgenii Kashin were supported by the EPSRC CDT in Intelligent Games & Games Intelligence (IGGI) (EP/S022325/1).

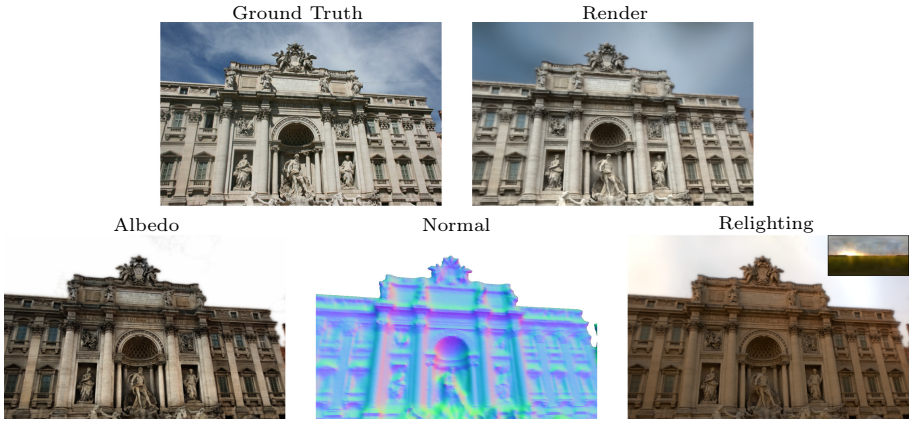

Fig. 12: Results on the Trevi Fountain scene showing decomposition and relighting.

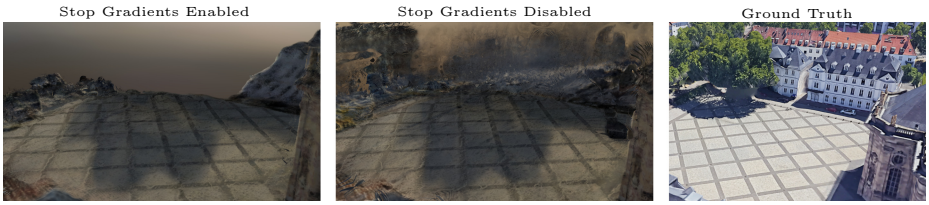

Fig. 13: With stop gradients enabled (left), geometry outside the view frustum of all training cameras is not generated. With stop gradients disabled (middle), gradients from appearance losses are allowed to flow from our sky visibility network to the SDF creating geometry not directly observed during training. This more closely matches the ground truth for the scene (right).

Fig. 14: Whilst the majority of the training images for *Site 3* in the NeRF-OSR dataset [30] show the front of the building in shadow. With our visibility network enabled our predicted albedo removes that shading along with shadows around the sign *(a)*, at the joint between brick and plaster *(b)* and on the ground *(c)*. Smaller cutouts show renderings with sky visibility on the left and without sky visibility on the right.

References

1. Aumentado-Armstrong, T., Tsogkas, S., Dickinson, S., Jepson, A.D.: Representing 3D Shapes With Probabilistic Directed Distance Fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19343–19354 (Jun 2022)
2. Barron, J.T., Malik, J.: Shape, Illumination, and Reflectance from Shading. TPAMI (2015)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. CVPR (2022)
4. Bell, S., Bala, K., Snavely, N.: Intrinsic images in the wild. ACM Transactions on Graphics (TOG) **33**(4), 1–12 (2014)
5. Boss, M., Jampani, V., Braun, R., Liu, C., Barron, J.T., Lensch, H.: Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=fATZNtA1-V0
6. Boss, M., Jampani, V., Kim, K., Lensch, H., Kautz, J.: Two-shot spatially-varying brdf and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3982–3991 (2020)
7. Bousseau, A., Paris, S., Durand, F.: User-assisted intrinsic images. In: ACM SIGGRAPH Asia 2009 papers, pp. 1–10. Association for Computing Machinery, New York, NY, United States (2009)
8. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: Pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5799–5809 (Jun 2021)
9. Chen, Z., Wang, G., Liu, Z.: Text2Light: Zero-Shot Text-Driven HDR Panorama Generation. ACM Trans. Graph. **41**(6) (Nov 2022). https://doi.org/10.1145/3550454.3555447, https://doi.org/10.1145/3550454.3555447, number of pages: 16 Place: New York, NY, USA Publisher: Association for Computing Machinery tex.articleno: 195 tex.issue_date: December 2022
10. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision Transformer Adapter for Dense Predictions. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=plKu2GByCNW
11. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
12. Dastjerdi, M.R.K., Hold-Geoffroy, Y., Eisenmann, J., Lalonde, J.F.: EverLight: Indoor-Outdoor Editable HDR Lighting Estimation (2023), arXiv: 2304.13207 [cs.CV]
13. Dave, A., Zhao, Y., Veeraraghavan, A.: PANDORA: Polarization-Aided Neural Decomposition of Radiance. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part VII. Lecture Notes in Computer Science, vol. 13667, pp. 538–556. Springer (2022). https://doi.org/10.1007/978-3-031-20071-7_32, https://doi.org/10.1007/978-3-031-20071-7_32, tex.bibsource: dblp computer science bibliography, https://dblp.org tex.biburl: https://dblp.org/rec/conf/eccv/DaveZV22.bib tex.timestamp: Mon, 05 Dec 2022 13:35:31 +0100

14. Dror, R.O., Willsky, A.S., Adelson, E.H.: Statistical characterization of real-world illumination. Journal of Vision **4**(9), 11–11 (Sep 2004)
15. Gardner, J.A.D., Egger, B., Smith, W.A.P.: Rotation-Equivariant Conditional Spherical Neural Fields for Learning a Natural Illumination Prior. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022), https://openreview.net/forum?id=cj6K4IWVomU
16. Gardner, J.A.D., Egger, B., Smith, W.A.P.: Reni++ a rotation-equivariant, scale-invariant, natural illumination prior (2023)
17. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 2335–2342. IEEE (2009)
18. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980, tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Thu, 25 Jul 2019 14:25:37 +0200
19. Kovacs, B., Bell, S., Snavely, N., Bala, K.: Shading annotations in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6998–7007 (2017)
20. Li, Q., Guo, J., Fei, Y., Li, F., Guo, Y.: NeuLighting: Neural Lighting for Free Viewpoint Outdoor Scene Relighting with Unconstrained Photo Collections. In: SIGGRAPH Asia 2022 Conference Papers. SA '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3550469.3555384, https://doi.org/10.1145/3550469.3555384, number of pages: 9 Place: Daegu, Republic of Korea tex.articleno: 13
21. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-Fidelity Neural Surface Reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
22. Liu, Z., Yang, B., Luximon, Y., Kumar, A., Li, J.: RayDF: Neural Ray-surface Distance Fields with Multi-view Consistency. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), https://openreview.net/forum?id=crZlhMnfe0
23. Loshchilov, I., Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=Skq89Scxx
24. Lyu, L., Habermann, M., Liu, L., Tewari, A., Theobalt, C., et al.: Efficient and differentiable shadow computation for inverse problems. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13107–13116 (2021)
25. Lyu, L., Tewari, A., Habermann, M., Saito, S., Zollhöfer, M., Leimküehler, T., Theobalt, C.: Diffusion posterior illumination for ambiguity-aware inverse rendering. ACM Transactions on Graphics **42**(6) (2023)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: ECCV (2020)
27. Müller, T., Evans, A., Schied, C., Keller, A.: Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, https://doi.org/10.1145/3528223.3530127, number of pages: 15 Place: New York, NY, USA Publisher: ACM tex.articleno: 102 tex.issue_date: July 2022

28. Pinzón, C., Jung, K.: Fast Python sampler for the von Mises Fisher distribution (Aug 2023), https://hal.science/hal-04004568, tex.hal_id: hal-04004568 tex.hal_version: v3
29. Rhodin, H., Robertini, N., Richardt, C., Seidel, H.P., Theobalt, C.: A versatile scene model with differentiable visibility applied to generative pose estimation. In: Proceedings of the 2015 International Conference on Computer Vision (ICCV 2015) (2015), http://gvv.mpi-inf.mpg.de/projects/DiffVis
30. Rudnev, V., Elgharib, M., Smith, W., Liu, L., Golyanik, V., Theobalt, C.: NeRF for Outdoor Scene Relighting. In: European Conference on Computer Vision (ECCV) (2022)
31. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
32. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)
33. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit Neural Representations with Periodic Activation Functions. In: Proc. NeurIPS (2020)
34. Somanath, G., Kurz, D.: Hdr environment map estimation for real-time augmented reality. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11298–11306 (June 2021)
35. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7495–7504 (Jun 2021)
36. Sun, J.M., Wu, T., Yang, Y.L., Lai, Y.K., Gao, L.: SOL-NeRF: Sunlight Modeling for Outdoor Scene Decomposition and Relighting. In: SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers '23) (2023)
37. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Kerr, J., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., McAllister, D., Kanazawa, A.: Nerfstudio: A Modular Framework for Neural Radiance Field Development. arXiv preprint arXiv:2302.04264 (2023)
38. Ueda, I., Fukuhara, Y., Kataoka, H., Aizawa, H., Shishido, H., Kitahara, I.: Neural Density-Distance Fields. In: Proceedings of the European Conference on Computer Vision (2022)
39. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=D7bPRxNt_AP
40. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=D7bPRxNt_AP
41. Wang, Z., Shen, T., Gao, J., Huang, S., Munkberg, J., Hasselgren, J., Gojcic, Z., Chen, W., Fidler, S.: Neural Fields meet Explicit Geometric Representations for Inverse Rendering of Urban Scenes. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2023)

42. Williams, L.: Casting curved shadows on curved surfaces. In: Proceedings of the 5th annual conference on Computer graphics and interactive techniques. pp. 270–274 (1978)
43. Worchel, M., Alexa, M.: Differentiable shadow mapping for efficient inverse graphics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 142–153 (2023)
44. Yao, Y., Zhang, J., Liu, J., Qu, Y., Fang, T., McKinnon, D., Tsin, Y., Quan, L.: NeILF: Neural Incident Light Field for Physically-based Material Estimation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 700–716. Springer Nature Switzerland, Cham (2022)
45. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. In: Thirty-Fifth Conference on Neural Information Processing Systems (2021)
46. Yenamandra, T., Tewari, A., Yang, N., Bernard, F., Theobalt, C., Cremers, D.: FIRe: Fast Inverse Rendering using Directional and Signed Distance Functions (2022), arXiv: 2203.16284 [cs.CV]
47. Yu, Y., Smith, W.A.P.: InverseRenderNet: Learning Single Image Inverse Rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2019)
48. Yu, Z., Chen, A., Antic, B., Peng, S.P., Bhattacharyya, A., Niemeyer, M., Tang, S., Sattler, T., Geiger, A.: SDFStudio: A Unified Framework for Surface Reconstruction (2022), https://github.com/autonomousvision/sdfstudio
49. Zhang, K., Luan, F., Wang, Q., Bala, K., Snavely, N.: Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5453–5462 (2021)
50. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: NeRFactor: Neural Factorization of Shape and Reflectance under an Unknown Illumination. ACM Trans. Graph. **40**(6) (Dec 2021). https://doi.org/10.1145/3478513.3480496, https://doi.org/10.1145/3478513.3480496, number of pages: 18 Place: New York, NY, USA Publisher: Association for Computing Machinery tex.articleno: 237 tex.issue_date: December 2021
51. Zins, P., Xu, Y., Boyer, E., Wuhrer, S., Tung, T.: Multi-View Reconstruction using Signed Ray Distance Functions (SRDF) (2023), arXiv: 2209.00082 [cs.CV]
52. Zobeidi, E., Atanasov, N.: A Deep Signed Directional Distance Function for Object Shape Representation. CoRR **abs/2107.11024** (2021), https://arxiv.org/abs/2107.11024, arXiv: 2107.11024 tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Fri, 04 Aug 2023 08:25:46 +0200