Objects as Cameras: Estimating High-Frequency Illumination from Shadows

Tristan Swedish, Connor Henley, and Ramesh Raskar

Massachusetts Institute of Technology
Cambridge, MA, 02139, USA
{tswedish, co24401, raskar}@mit.edu

Abstract

We recover high-frequency information encoded in the shadows cast by an object to estimate a hemispherical photograph from the viewpoint of the object, effectively turning objects into cameras. Estimating environment maps is useful for advanced image editing tasks such as relighting, object insertion or removal, and material parameter estimation. Because the problem is ill-posed, recent works in illumination recovery have tackled the problem of low-frequency lighting for object insertion, rely upon specular surface materials, or make use of data-driven methods that are susceptible to hallucination without physically plausible constraints. We incorporate an optimization scheme to update scene parameters that could enable practical capture of real-world scenes. Furthermore, we develop a methodology for evaluating expected recovery performance for different types and shapes of objects.

1. Introduction

Consider a small object sitting on a desk in your living room. The object is illuminated by light sources from all directions—this includes direct sources such as the sun or overhead lights, but also indirect sources, like the foliage outside that scatters sunlight through your window. The appearance of the object and the surface that it rests upon results from the complex interaction between the incident illumination and the geometry and material properties of the object and the desk. In this paper we ask the question—if the geometry and material properties of the observed scene are known, how well can we reconstruct the incident illumination pattern?

If we assume that the illumination sources are distant relative to the size of the observed object, then we can represent this illumination as a hemispherical photograph taken from the perspective of the object. Thus, by using our knowledge about an object to accurately estimate the illumination incident upon it, we effectively turn the object into a camera.

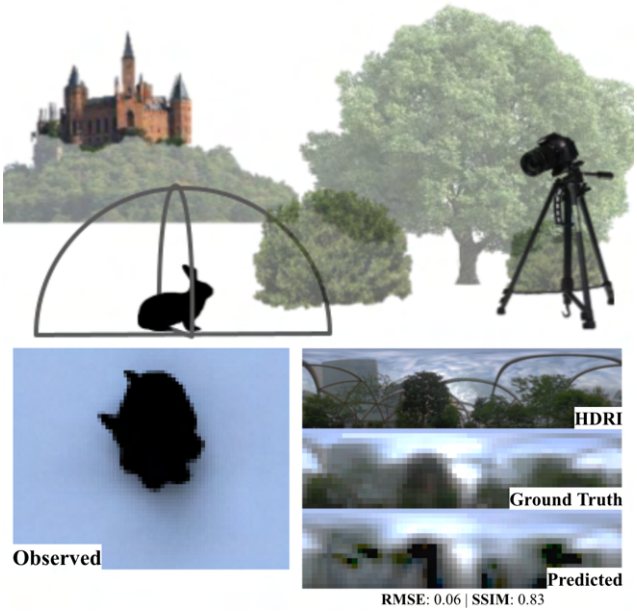

Figure 1. (Top) Observing the area around a small, bunny-shaped object (top-left), can we recover occluded viewpoints only visible from the bunny's perspective? (Bottom) Given the surface geometry of an object (bottom-left), we estimate the incident illumination, and in some cases, the unknown diffuse albedo of the surface surrounding the object.

In this paper, we primarily make use of shadows cast by an object onto nearby surfaces. Cast shadows are particularly easy to interpret when an object is illuminated from a single direction. For example, one can immediately determine the position of the sun by looking at a sundial. Estimating the illumination incident from all directions simultaneously is more challenging, and is a linear but ill-posed inverse problem.

Prior methods are commonly limited to controlled capture methodologies. Some rely upon object's shading or specular highlights and thus require the bidirectional reflectance distribution function (BRDF) or precise surface normal information. Our method relies on cast shadows and requires only the shape of the object and the shape and albedo of the shadowed surface. Previous works that utilize cast shadows require planar objects that cast shadows onto planar surfaces. We aim to support complex 3D objects in a general framework applicable to most natural objects and scenes.

We require partial reconstruction of the visible scene surface, such as from a stereo camera pair, or utilize known object geometry and relative camera orientation. Our estimates are performed by jointly optimizing for scene parameters using this partial set of views to approximate the ray transport matrix. Once we have estimated the ray transport matrix, further estimates only require a single photograph of the scene.

1.1. Contributions

- Recover high-frequency environment map from object shadows that does not require a special setup, specular object, or low-frequency assumptions
- Propose a practical technique for approximating the light transport of an arbitrary object in a natural setting from a restricted set of partial viewpoints
- Demonstrate a strategy for solving the inverse problem "in the wild" for a scene with unknown surface material and using just a single photograph.
- Examine the structure of the ray transport matrix to determine feasible regions of reconstruction and assess the object-camera performance

2. Related Work

2.1. Incident Illumination Estimation

Environment map estimation techniques usually require specific capture setups [13], or seek approximations useful for downstream tasks [21, 44]. Sato et al [34] were one of the first to define the problem of approximating illumination from shadows of objects on diffuse surfaces. Later work has emphasized the use of sharp shadow boundaries from a few bright sources for mixed reality applications or relighting, [17, 26, 14, 46] rather than extended sources. Other approaches include the use of custom probes for recovering lighting useful for scene shading [8, 7].

Recently, Jiddi et al [18] demonstrated illumination estimation using both specular paths and shadow information. Specular paths are a useful cue for estimating incident illumination [16, 31] potentially "in the wild," but suitable surfaces must be present in the scene.

Data-driven techniques have been used to estimate incident illumination for a database of objects [43]. Large datasets have been used to train deep neural networks to predict incident illumination in natural scenes [15, 23] for augmented reality applications or learning illumination for portrait relighting [24]. Spatially varying illumination estimation has been demonstrated using RGBD images [5] and photographs [15], but these models do not enforce physical plausibility.

Our work builds on existing methods for estimating illumination from shadows, with a particular focus on arbitrary illumination conditions, such as extended sources, and scenes with unknown albedo.

2.2. Inverse Rendering

Recently, several data-driven approaches have been proposed to learn a mapping between reference photographs and scene parameters [32] [2, 9] directly. These learned mappings typically incorporate a differentiable rendering module to serve as conditioning during training [2, 9, 19, 40].

Volumetric scene representations have been proposed for spatially varying lighting estimation [38]. Similarly, implicit neural representations have shown impressive results for inverse rendering tasks [28, 37], but estimates the unmixed product of material reflectance and lighting.

Recently, practical implementations of differentiable path tracing for the rendering of mesh-parameterized geometries have been developed [25], as well as a reparameterization that makes use of modern autodifferentiation techniques [30, 27].

Demonstrations of de-rendering have typically been used for re-lighting or 3D reconstruction tasks and are less focused on "image-like" illumination estimates.

2.3. Occlusion Assisted Imaging

Early occlusion-based non-line-of-sight (NLoS) approaches make use of specific scene features such as accidental cameras or pinspecks [11, 42]. More recently, some different capture methodologies have been proposed for time-of-flight based approaches [33], blind deconvolution in intensity-based NLoS [29, 35, 45, 41], and the recovery of light fields [4]. Critically, these works have shown reconstructions from calibrated planar surfaces. The use of the deep image prior has been proposed to approximate the ray transfer matrix [1], but assumed planar hidden scenes and did not investigate realistic high dynamic range hidden scenes.

Bouman and Seidel [6, 36] each demonstrated how the occlusion of light at a flat edge enables the reconstruction of 1D projected views of a scene hidden around a corner. Other approaches have utilized active sources to illuminate hidden objects [20, 39, 10]. Visual deprojection [3]

is a learning-based approach to estimate 2D scenes from 1D projections, and like other learning-based approaches, is limited to specific scene setups and has not been shown to work on novel and diverse scenes. Our method supports the use of objects with arbitrary shape, and does not require planar or one-dimensional masks or planar, calibrated relay surfaces.

3. The Object Camera

A perspective camera separates the radiance of light rays that travel through a particular point in space. A pinhole camera's aperture separates rays such that each sensor position receives light from a unique direction. In our proposed setting, a chosen object forms the camera's "aperture" and the surrounding visible surface forms the sensor. Given an image of this object and the shadows that it casts, we hope to "see" what the object can see from its own perspective.

3.1. Inverse Rendering Problem

Inverse rendering is an analysis by synthesis approach, where we optimize the parameters of a forward light transport model until the rendered images closely resemble the measured image. In general, the inverse rendering problem is extremely ill-posed: there are some combinations of different scene parameters that render very similar looking images. As such, additional information is required in the form of priors or additional images to resolve ambiguities.

To illustrate the problem, consider a scene with a single object on a flat surface that has an arbitrary albedo and diffuse surface reflectance. We aim to reconstruct an image from the object's perspective, represented by the set of rays extending from the center of the object to a distant hemisphere. If we consider only direct illumination, the observed radiance from a scene point, x, can be written as follows:

$$L_o(x, \omega_o) = \rho(x) \int_\Omega V(x, \omega_i) L_i(\omega_i)(\omega_i \cdot \mathbf{n}) d\omega_i, \quad (1)$$

where L_o is the radiance observed by the camera at surface position x with surface normal $\mathbf{n}$, ρ is the spatially varying diffuse albedo, V is the visibility of the surrounding hemisphere, parameterized by ω_i, as seen from surface point x, and L_i is the radiance of every incident ray from the surrounding hemisphere. We note that L_i forms an image of the incident illumination from the perspective of the object. We assume that the illuminating hemisphere is sufficiently far away that we only need to know ω_i. This illumination model is often referred to as an environment map.

If ρ and V are known, solving for L_i is a linear inverse problem. Concretely, if we discretize these spatially varying functions, with incident illumination vector $\mathbf{x}$ corresponding to illumination directions sampled over the environment map, and $\mathbf{y}$ the pixels of the corresponding image, we can write the above equation in matrix form, where "$\odot$" is the Hadamard product:

$$\mathbf{y} = \text{diag}(\boldsymbol{\rho})(\mathbf{V} \odot \mathbf{C})\mathbf{x} = \mathbf{A}\mathbf{x}, \quad (2)$$

Where $\boldsymbol{\rho}$ is a $M \times 1$ vector of diffuse albedos, $\mathbf{V}$ is a $M \times N$ matrix, $\mathbf{C}$ is a $M \times N$ matrix of the cosine factors, and $\mathbf{x}$ is a $N \times 1$ vector of unknown illumination radiance. Combining factors, we obtain a linear system of equations represented by the matrix $\mathbf{A}$, where $\mathbf{A} = \text{diag}(\boldsymbol{\rho})(\mathbf{V} \odot \mathbf{C})$. We can therefore solve for the environment map, $\mathbf{x}$, by minimizing the mean squared error: $||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2$.

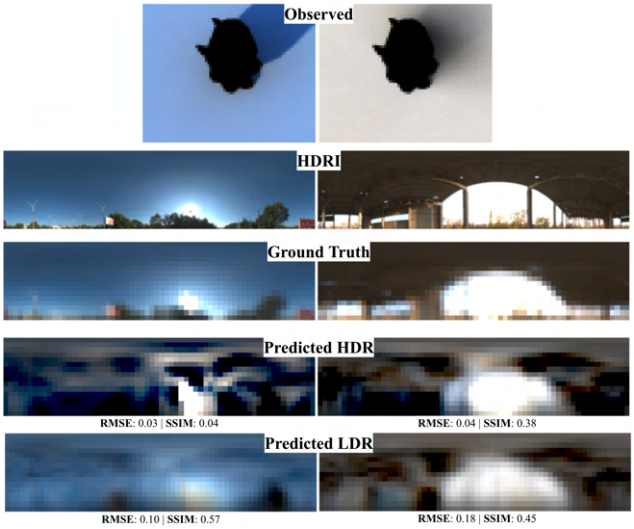

Figure 2. Given a known surface geometry and shadow surface albedo, we estimate the environment map illumination using a single observed image by solving Eq. 3. For easier viewing, basic tonemapping was applied to all images using gamma correction with clipping ($\gamma = 2.2$, image normalized such that the sun is clipped).

3.2. Solving for Incident Illumination

Overview of Approach Rather than decompose the matrix explicitly for every scene as in Eq. 2, we make use of a modern rendering framework to represent the more complex structure of the model matrix, $\mathbf{A}$. Since we model the incident illumination using an environment map, we can generate the rows of $\mathbf{A}$ by rendering an image for each illumination source in the discretized environment map.

Our inverse rendering problem is composed of two steps:

1. Render the approximate light transport matrix using estimated scene parameters
2. Estimate illumination or albedo by solving the corresponding linear system

The scene parameters we estimate at step 1 are the camera extrinsic parameters, scene surface mesh, and diffuse

surface albedo. These parameters could be known *a priori*, or obtained using some other 3D reconstruction methodology such as photogrammetry.

Rendering the Ray Transport Matrix We utilize the Mitsuba 2 [30, 27] path tracer to render the ray transport matrix. Each column in the $\mathbf{A}$ matrix is associated to an individual pixel in our environment map, and consists of the image that is rendered when only that single environment map pixel is lit. The rendered images are the columns of $\mathbf{A}$: $\mathbf{A_i} = f_\theta(\mathbf{x_i})$ where $\mathbf{x_i}[i] = 1$ and $\mathbf{x_i}[j] = 0$ when $i \neq j$. $f_\theta(\mathbf{x})$ represents the Mitsuba 2 rendering pipeline given scene parameters θ. In our case, θ are the surface geometry mesh data and uv texture maps with corresponding diffuse albedo or other material parameters. We map $\mathbf{x}$ to the texture data used by the environment map emitter, where image pixels correspond to latitude and longitude in spherical coordinates.

Jointly Solving for Illumination and Albedo With known surface geometry and material parameters, we can solve for the incident illumination. We write the illumination recovery problem as the solution to a linear system, where the rows of the model matrix, $\mathbf{A}$, are comprised of rendered images for each component (e.g. ray) of the illumination model. Since $\mathbf{A}^\top\mathbf{A}$ is not necessarily well conditioned, we add spatial smoothness and L2 regularization, such that we aim to minimize the following objective:

$$\begin{aligned} \arg\min_{\mathbf{x}} \quad & ||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2 + \lambda_s||\mathbf{D}\mathbf{x}||_2^2 + \lambda_r||\mathbf{x}||_2^2 \\ \text{s.t.} \quad & \mathbf{x} >= 0, \end{aligned} \tag{3}$$

where $\mathbf{D}$ is the spatial difference operator and λ_s, λ_r are regularization parameters controlling spatial smoothness and L2 regularization respectively.

Typically, the estimated albedo of the shadow surface is unavailable or not accurate enough to obtain good results. We found that jointly estimating albedo and illumination was essential to make our model robust to poor initialization of mesh albedo, due to occasional artifacts in the surface texture produced by photogrammetry. Given a ray transfer matrix, $\mathbf{A}'$, rendered using Mitsuba with or without an existing albedo texture map, we augment our forward model with a per-pixel scaling factor as a proxy for diffuse albedo: $\mathbf{A} = \text{diag}(\rho)\mathbf{A}'$.

As pointed out in [36], we also observed that adding a pixel-wise scaling term to emulate the diffuse albedo residuals seemed to make the illumination estimates more robust. As such, given the ray transport matrix rendered using Mitsuba, $\mathbf{A}'$, we can solve for the per-pixel scaling term. Since the albedo term is fixed under changing illumination conditions, it is advantageous to write the following objective for L observed images, $\mathbf{y_l}$, and the associated environment map, $\mathbf{x_l}$.

Thus, we minimize the following objective:

$$\begin{aligned} \arg\min_{\boldsymbol{\rho}} \quad & \sum_l ||\text{diag}(\mathbf{A}'\mathbf{x_l})\boldsymbol{\rho} - \mathbf{y_l}||_2^2 + \lambda_p||\mathbf{G}\rho||_2^2 \\ \text{s.t.} \quad & 0 < \boldsymbol{\rho} \leq 1, \end{aligned} \tag{4}$$

where $\mathbf{G}$ is the linear inverse operator: $\mathbf{G} = ({\mathbf{A}'}^\top\mathbf{A}' + \lambda_r\mathbf{I})^{-1}{\mathbf{A}'}^T$. This is similar to the regularization term used in [36], and discourages albedo estimates that can be easily reconstructed using the ray transport matrix, such as cast shadows.

In practice, we solve for the albedo term, $\boldsymbol{\rho}$, and each environment map, $\mathbf{x_l}$, separately. We solve Equations 3 and 4 using damped Newton's method, with update steps in the direction of $\mathbf{x}^*$ and $\boldsymbol{\rho}^*$, defined below.

The unconstrained minimum solution of Equation 3, $\mathbf{x}^*$, satisfies the linear equation, with $\mathbf{R} = \lambda_s\mathbf{D}^\top\mathbf{D} + \lambda_r\mathbf{I}$:

$$({\mathbf{A}'}^\top\text{diag}(\boldsymbol{\rho})^2\mathbf{A}' + \mathbf{R})\mathbf{x}^* = (\text{diag}(\boldsymbol{\rho})) \odot \mathbf{A}')^\top\mathbf{y} \tag{5}$$

Similarly for the $\boldsymbol{\rho}^*$ and Equation 4:

$$(\lambda_p\mathbf{G}^\top\mathbf{G} + \sum_l \text{diag}(\mathbf{A}'\mathbf{x_l})^2)\rho^* = \mathbf{y_l} \odot \sum_l \mathbf{A}'\mathbf{x_l} \tag{6}$$

Equations 5 and 6 can be solved efficiently using a linear solver. With $\mathbf{x}^*$ and $\boldsymbol{\rho}^*$, we alternately update $\boldsymbol{\rho}^{t+1}$ and $\mathbf{x}^{t+1}$ until convergence using a damped Newton step with damping factor γ, applying a projection H after each step:

$$\begin{aligned} \boldsymbol{\rho}^{t+1} =& H_{0,1}(\boldsymbol{\rho}^t + \gamma(\boldsymbol{\rho}^* - \boldsymbol{\rho}^t)) \\ \mathbf{x}^{t+1} =& H_{0,\infty}(\mathbf{x}^t + \gamma(\mathbf{x}^* - \mathbf{x}^t)) \end{aligned} \tag{7}$$

Where H is a simple clipping operator, with $\min \leq H_{\min,\max}(\cdot) \leq \max$.

Combining Multiple Images While this work evaluates data from a single camera pose, our framework can be applied to P camera poses each with L illumination conditions as described below.

Multi-Illumination For a static camera with L different illumination conditions, we solve for each $\mathbf{x_l}^*$ environment map separately. Updating the albedo scaling factor, $\boldsymbol{\rho}$, only requires one solution once each environment map, $\mathbf{x_l}$, is computed.

Multi-Viewpoint If there are P viewpoints, we find $\mathbf{x}^*$ by solving a larger system of equations, by stacking the light transport matrix associated with each camera pose, $\mathbf{A_p}$, and observed images $\mathbf{y_p}$. We estimate a single environment map while incorporating information from all viewpoints. To find $\rho_\mathbf{p}$ for each camera, the set of $\mathbf{y_p}, \mathbf{A_p}$ are used to solve each $\rho_\mathbf{p}{}^*$ separately.

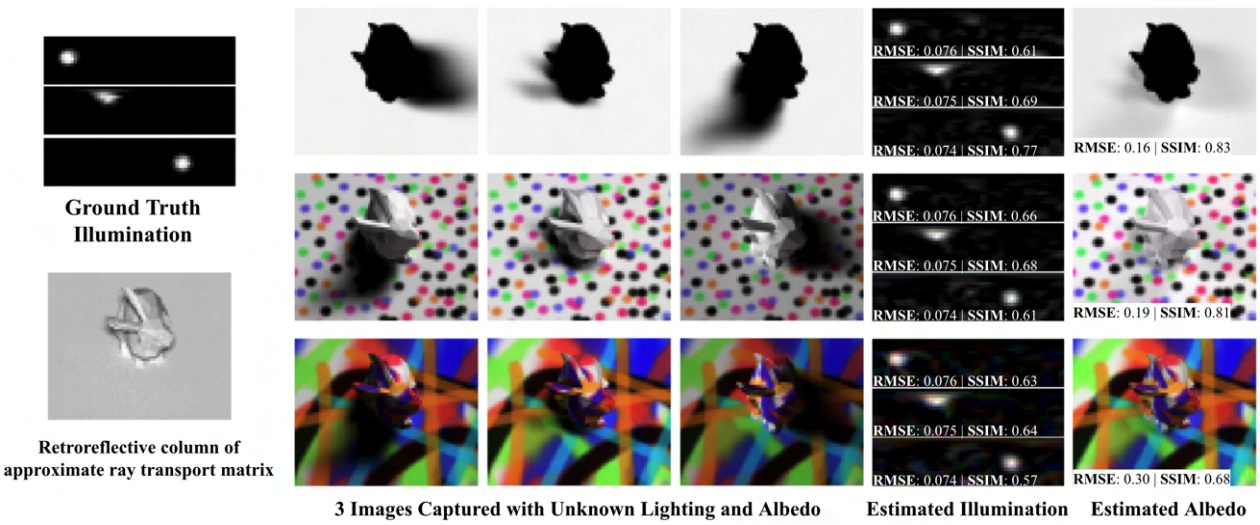

Figure 3. Given a ray transport matrix computed assuming all surface albedo are one, and 3 images with unknown surface albedo or illumination, we estimate both using the update procedure described in Equation 7. Three different texture maps are shown, the observed images are rendered with global illumination.

4. Implementation

Image Capture For all of our experiments, we capture a single "target image", $\mathbf{y}$, using a DSLR camera (Canon EOS Rebel T5). For alignment to the object, we use a photogrammetry pipeline that includes structure from motion, mesh generation, and surface albedo estimation. In some of our real experiments, we use the known geometry. For the ground truth environment map, we photograph a chrome ball with exposure bracketing for high-dynamic-range (HDR).

Rendering System For all experiments, we performed the rendering computations on a single Nvidia RTX 2080. Ray transport matrices took between 200-300 seconds (scene dependent) to render 2048 total images with environment map textures sized 32×64. Results in Fig. 3 converged in about 5 iterations ($\sim$25 secs/iter) running on the CPU.

4.1. Synthetic Results

Realistic HDR Environments We rendered the ray transport matrix with known geometry and albedo for the black albedo bunny supported by a white planar surface. Each row of the ray transport matrix was rendered using 32 bit floats with 768 samples per pixel (spp). Observed images were rendered with 25.6k spp using freely available high dynamic range (HDR) probe images of real environments.[1]. We show the effect of renders with fewer spp and larger variance in the supplement, as well as an analysis of additive noise.

To solve Equation 5, each ray transport matrix requires approximately 500 MB of memory. We found that reconstructions took a few seconds on our machine using an Intel Xeon Silver 4210 CPU with 256GB RAM. As we see in Figure 2, we achieve reconstructions of the HDR environment maps with a reasonable recovery of the relative intensity of the bright sources.

Unknown Albedo and Illumination Conditions In practical scenarios, the surface albedo of an object may be unknown. We can overcome poor estimates of the surface albedo by jointly recovering the albedo and illumination. In Figure 3, we separate the albedo and illumination for three different textured versions of the bunny under a bright moving illumination source. Our initial ray transport matrix is generated using an all-white albedo for both the bunny and shadow surface. Upon initializing the albedo and environment map to all ones, we apply the update scheme in Equation 7 with $\gamma = 0.9$ for 20 iterations. We notice that the first update quickly resolves the albedo, but it often takes an additional iteration before the environment maps begin to converge. While both Equation 5 and Equation 6 can be solved in one step, we found it useful to reduce the learning rate such that the projection operator could enforce the constraints without collapsing the environment map solution to all zeros.

Failure Cases We found that HDR environment maps with extreme dynamic range can pose a challenge for reconstruction. In Figure 2, the bright sun has a radiance of $\sim 10000:1$ compared to the sky. Our approach attempts to estimate the HDR scene, but primarily resolves the brightest sources and ignores the detail in darker areas of the scene.

[1] https://hdrihaven.com

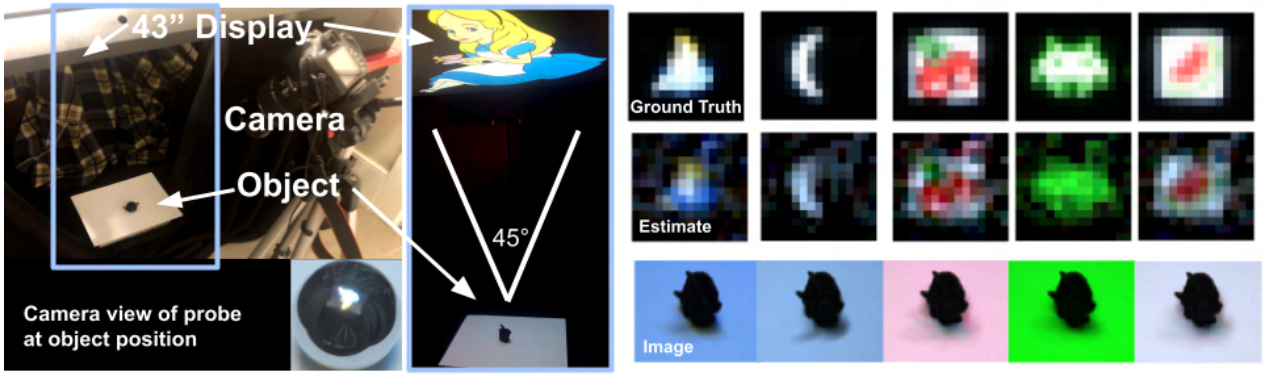

Figure 4. An object of known shape (a bunny) is placed in a controlled illumination environment. Illumination patterns are displayed on an LCD display placed above the object, and a camera observes the shadows that the object casts onto a flat surface. From this single image we produce an estimate of the illumination pattern seen by the object. A chrome ball is used to collect ground truth data in the object's position for each test pattern.

When the environment map is clipped, producing a low-dynamic range (LDR) scene (only possible in simulation), our method is able to recover finer details. While we did not investigate further in this work, encouraging the recovery of darker regions is challenging because any HDR compression added to the output of the forward model would make it nonlinear. Similarly, since our method relies on tiny variations in the observed scene, our method is susceptible to errors in the HDR capture process.

The recovered albedo maps contain minor artifacts resembling the shadows in the observed images. We unsuccessfully applied a wide range of values for the regularization proposed in [36]. However, diverse and extended sources help reduce the appearance of these artifacts, but that may be due in part to the softer shadows cast by these objects. We were able to estimate convincing albedo using environment maps of real scenes, but the associated estimated environment maps often contain "retro-reflection" artifacts along the retro-reflective direction from the camera to object. These artifacts only appear after the first few iterations of projected gradient descent, and so reasonable results are achieved using early stopping. We show additional results in the supplement highlighting these failure modes.

Gradient Descent Baseline We compared our method in Table 1 to a stochastic gradient descent (SGD) baseline by minimizing reconstruction error using Mitsuba 2's auto-differentiation capabilities. We update both the scene albedo and environment map using Mitsuba 2's Adam optimizer, with learning rate 0.01. For the multi-illumination/unknown albedo scenes, the recovered albedo using Mitsuba 2+Adam was slightly better in terms of RMSE, but remained quite noisy, reducing SSIM. We believe Mitsuba 2's improved albedo estimation is due to the fact that our linear diffuse albedo model does not account for global illumination. As such, applying our proposed method to quickly solve the linear approximation and then fine-tuning for non-linear effects using a differentiable renderer is a promising direction for future work.

Our approach has a number of advantages over gradient descent. In the simpler case where scene albedo is known, the loss function in Eq. 3 can be solved exactly in a single step using a linear solver ($\sim$4.5 secs in our CPU numpy implementation). Additionally, once the albedo has been estimated, the regularized inverse, $\mathbf{G}$, can be applied to new measurements to quickly estimate novel illumination without re-computing $\mathbf{A}$. Furthermore, our analysis in Section 5 is made possible by first rendering $\mathbf{A}$. We show in Table 1 that our method converges faster than Adam+Mitsuba 2. However, since our method effectively computes the Hessian, $\mathbf{A}^\top\mathbf{A}$, this advantage may not scale to high resolution environment maps. Regardless, faster low resolution estimates, like the proposed method, would remain useful for initialization.

4.2. Real-data Results

3D Printed Probes We 3D printed a 5cm tall "Utah Teapot" and low poly "Stanford Bunny" in black filament such that the 3D shape of the object is known, but the surface of the object would make it difficult to use any other cues for incident illumination estimation.

Each object was placed below an LCD panel (43" LG Desktop Monitor), on a white sheet of computer paper, approximately one meter away from the display. The shortest and longest monitor dimensions correspond to a 45 and 90 degree field of view from the perspective of the object respectively. The display was used to show different patterns, and an image of the object was captured with a known camera orientation, about one meter from the 3D printed object.

Scene / Method	Envmap RMSE	Envmap SSIM	Albedo RMSE	Albedo SSIM	Compute Time (s)
Multi-illumination "dots"	0.417	0.43	0.286	0.78	**327**
+L2	**0.075**	0.62	0.219	**0.81**	**327**
+L2+Smooth (ours)	**0.075**	0.63	0.219	**0.81**	**327**
+L2+Smooth+[36]	**0.075**	0.63	0.218	**0.81**	338
Grad Descent Baseline (early stop)	0.531	0.41	0.202	0.69	**327**
Grad Descent	0.501	**0.68**	**0.172**	0.73	1434
Known albedo "garden"	0.492	0.24	-	-	**225**
+L2	0.078	0.77	-	-	**225**
+L2+Smooth (ours)	**0.063**	**0.83**	-	-	**225**
Grad Descent Baseline (early stop)	0.294	0.31	-	-	**225**
Grad Descent	0.120	0.57	-	-	975

Table 1. Ablation study: **Multi-illumination**: RMSE/SSIM relative to the known synthetic ground truth for the unknown albedo "dots" scene computed for 5 iterations for each set of added regularization terms. **Known albedo**: RMSE/SSIM computed for the HDRI "garden" scene for a single iteration. The compute times include pre-rendering of the ray transport matrix (220 secs) for the proposed method. Gradient Descent was stopped early to compare results under similar compute times, corresponding to ∼3461 iterations. The learning rate for Adam, $\gamma = 0.01$, for the gradient descent baseline was chosen to maximize SSIM after full convergence, which took significantly more time than the proposed method (+L2+Smooth).

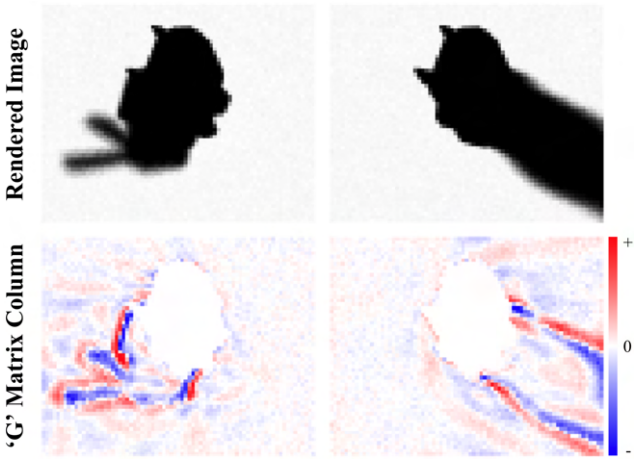

Figure 5. Our algorithm implicitly amplifies the edge gradients of shadows cast from a particular direction. (Top Row) Shadows rendered for point illumination above and to the right, and behind and to the left of the bunny-shaped occluder. (Bottom row) Corresponding rows of regularized inverse operator.

Reconstructions are shown for each test pattern in Figure 4.

Figure 4 shows cropped and centered environment maps with each side corresponding to a 79 degree field of view. There are some apparent distortions in the reconstructions, such as a missing corner in Figure 4(e). We also produced reconstructions using much higher environment map resolutions (up to 256 × 512) and higher input image resolutions, but found they did not produce significantly improved reconstructions while taking longer to compute.

5. Analysis

Extracting the Shadow's Edge In Figure 5 we show that our algorithm implicitly amplifies the edge gradients to determine the intensity of illumination from a particular direction. In the bottom row we plot the rows of the inverse operator matrix $\mathbf{G} = (\mathbf{A}^\top\mathbf{A} + \mathbf{R})^{-1}\mathbf{A}^\top$ (referred to in [6] as "estimation gain images"). These rows are correlated with the input image to estimate the intensity of illumination originating from a particular direction.

Determining Influential Image Pixels Given a target image measurement, we can quantify the influence that individual pixels have on our estimated environment map using a statistical tool known as Cook's distance [12]. The Cook's distance measures the effect that removing a measurement has on an estimated curve fit, and can be expressed as follows:

$$\mathbf{D_i} = \frac{\sum_{j=1}^{M}(\hat{y}_j - \hat{y}_{j(i)})^2}{ps^2} = \frac{e_i^2}{ps^2}\left[\frac{h_{ii}}{(1-h_{ii})^2}\right] \quad (8)$$

Here $\hat{y}_j$ denotes the re-projected curve fit (that is, the image rendered using our environment map estimate $\hat{x}$) obtained when all measurements are used for the fit, and $\hat{y}_{j(i)}$ denotes the fit obtained after the i^{th} data point has been removed from the measurement set. The quantity $s^2 = \frac{\mathbf{e}^T\mathbf{e}}{M-N}$ is the mean square error of the fit $\hat{\mathbf{y}}$ calculated from the residual vector $\mathbf{e} = (\mathbf{y} - \hat{\mathbf{y}})$ and the dimensions of the model matrix. The value h_{ii} is referred to as the *leverage* of the measurement y_i and is defined as the i^{th} diagonal element of the hat matrix $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top\mathbf{A} + \lambda_r\mathbf{I})^{-1}\mathbf{A}^\top$.

In Figure 6 we've plotted an image of Cook's Distance corresponding to a specific target image measurement. As expected, we notice that the pixels with the largest Cook's Distance appear to lie within the shadowed regions.

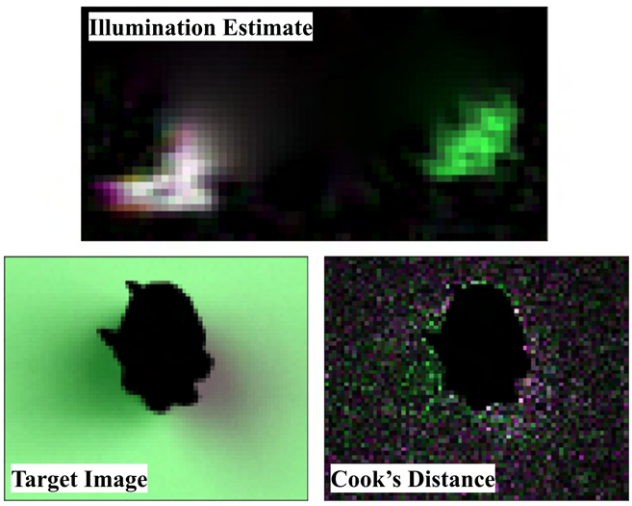

Figure 6. We plot an image of the Cook's distance (bottom right) associated with each pixel in the target image shown on the bottom left. The estimated environment map associated with this image is shown in the top row. The Cook's distance image has been compressed with a gamma value of 0.5 to highlight interesting features.

Assessing Object-Camera Performance Although the Cook's Distance is useful for determining which pixels in a specific set of measurements are most influential, we may also want to assess how influential individual pixels are in general, independent of any specific set of measurements. For this purpose, the *leverage* of individual measurement channels can be a useful metric. The leverage of pixel i, previously defined as the i^{th} diagonal component of the hat matrix, can also be defined as follows:

$$h_{ii} = \mathbf{a}_i^*(\mathbf{A}^\top\mathbf{A} + \lambda_r\mathbf{I})^{-1}\mathbf{a}_i^{*\top} = \frac{\partial\hat{y}_i}{\partial y_i} \tag{9}$$

Here $\mathbf{a}_i^*$ corresponds to the i^{th} row of the matrix A. Figure 7 includes an image of per-pixel leverage values calculated for the bunny-camera. We note that, as with Cook's distance, the pixels closest to the base of the bunny appear to be most influential.

Given a particular object-camera configuration, we might also be interested in assessing which entries in an environment map we can expect to reconstruct accurately. We take the square roots of the diagonal entries of the covariance matrix of our least-squares fit: $\mathbf{\Sigma} = (\mathbf{A}^\top\mathbf{A} + \lambda_r\mathbf{I})^{-1}$—that is, the inverse of the Hessian of the loss function defined in Eq. 3. We ignore the scene smoothness prior for the sake of analyzing the intrinsic properties of the object-camera.

A plot of these relative uncertainty values is also shown in Figure 7. From this image, we anticipate that the bunny-camera will be best at reconstructing illumination arriving from above and slightly to the left or right of the bunny.

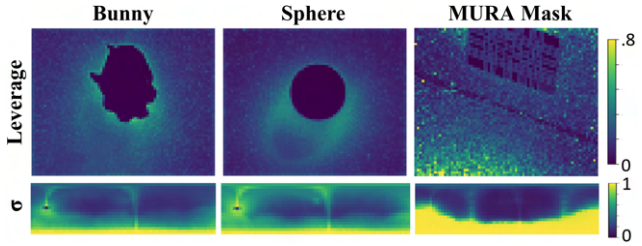

Figure 7. We illustrate how occluder shape impacts object-camera performance by generating images of per-pixel leverage (top row) and environment map uncertainties (bottom) for three occluder shapes: a bunny, a sphere, and a coded aperture mask [22].

Effect of Occluder Shape Our analysis of the bunny-camera makes it clear that the shape of the occluder can have a significant effect on the object camera's performance. This has important practical ramifications. For instance, we might choose to opportunistically exploit occluders found "in the wild" that are likely to produce accurate reconstructions of illumination originating from certain directions. Alternatively, we could design an optimized occluder shape that can be 3D printed and used as an object camera in the real world.

We demonstrate the effect of occluder shape in Figure 7. We show leverage and uncertainty maps for three different occluder shapes—a bunny, a sphere, and a 2D coded aperture mask. Compared to the bunny-camera, the sphere-camera achieves reconstruction uncertainties that are more uniform across the hemisphere, but that are higher on average. In contrast, the coded aperture mask achieves very low uncertainties when the shadow of the mask falls within the camera field of view, but uncertainty is high when the mask is illuminated edge-on, and very high when light originates near the horizon.

6. Conclusion

Reconstruction using the shadows cast by objects onto their surrounding surface can be practically achieved using tools used commonly in computer vision and graphics research. Shadows are an important cue for high-frequency incident illumination estimation and can be essential for solving an otherwise poorly-conditioned problem.

We hope this work will inspire further extensions, combining many exciting research directions in computer vision, from illumination estimation for image relighting and augmented reality, to imaging beyond and around the line of sight.

Acknowledgements We thank our reviewers for their helpful comments. This work was supported by DARPA REVEAL (N00014-18-1-2894) and the Media Lab Consortium. TS was supported in part by NSF GRFP (No. 1122374).

References

[1] Miika Aittala, Prafull Sharma, Lukas Murmann, Adam Yedidia, Gregory Wornell, Bill Freeman, and Fredo Durand. Computational mirrors: Blind inverse light transport by deep matrix factorization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 14311–14321. Curran Associates, Inc., 2019. 2

[2] Dejan Azinović, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 2

[3] Guha Balakrishnan, Adrian V. Dalca, Amy Zhao, John V. Guttag, Fredo Durand, and William T. Freeman. Visual deprojection: Probabilistic recovery of collapsed dimensions. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[4] M. Baradad, V. Ye, A. B. Yedidia, F. Durand, W. T. Freeman, G. W. Wornell, and A. Torralba. Inferring light fields from shadows. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6267–6275, June 2018. 2

[5] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, June 2013. 2

[6] Katherine L. Bouman, Vickie Ye, Adam B. Yedidia, Fredo Durand, Gregory W. Wornell, Antonio Torralba, and William T. Freeman. Turning corners into cameras: Principles and methods. In *ICCV*, 2017. 2, 7

[7] Dan Calian. *Customised Light Probes and Inverse Lighting Methods for Relighting*. PhD thesis, University College London, 2019. 2

[8] Dan A. Calian, Kenny Mitchell, Derek Nowrouzezahrai, and Jan Kautz. The shading probe: Fast appearance acquisition for mobile ar. In *SIGGRAPH Asia 2013 Technical Briefs*, SA '13, New York, NY, USA, 2013. Association for Computing Machinery. 2

[9] C. Che, F. Luan, S. Zhao, K. Bala, and I. Gkioulekas. Towards learning-based inverse subsurface scattering. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2020. 2

[10] W. Chen, S. Daneau, C. Brosseau, and F. Heide. Steady-state non-line-of-sight imaging. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6783–6792, June 2019. 2

[11] A. L. Cohen. Anti-pinhole imaging. *Journal of Modern Optics*, 29:63–67, 1982. 2

[12] R. Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18, 1977. 7

[13] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 189–198, New York, NY, USA, 1998. Association for Computing Machinery. 2

[14] Sylvain Duchêne, Clement Riant, Gaurav Chaurasia, Jorge Lopez Moreno, Pierre-Yves Laffont, Stefan Popov, Adrien Bousseau, and George Drettakis. Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Trans. Graph.*, 34(5), Nov. 2015. 2

[15] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-Francois Lalonde. Fast spatially-varying indoor lighting estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[16] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. What is around the camera? In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[17] S. Jiddi, P. Robert, and E. Marchand. Estimation of position and intensity of dynamic light sources using cast shadows on textured real surfaces. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1063–1067, 2018. 2

[18] Salma Jiddi, Philippe Robert, and Eric Marchand. Detecting specular reflections and cast shadows to estimate reflectance and illumination of dynamic indoor scenes. *IEEE transactions on visualization and computer graphics*, 2020. 2

[19] H. Kato, Deniz Beker, M. Morariu, T. Ando, T. Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *ArXiv*, abs/2006.12057, 2020. 2

[20] Jonathan Klein, Christoph Peters, Jaime Martín, Martin Laurenzis, and Matthias B. Hullin. Tracking objects outside the line of sight using 2D intensity images. *Sci. Rep.*, 2016. 2

[21] Jean-François Lalonde, Alexei A. Efros, and Srinivasa G. Narasimhan. Estimating the natural illumination conditions from a single outdoor image. *International Journal of Computer Vision*, 2011. 2

[22] Douglas Lanman, Ramesh Raskar, Amit Agrawal, and Gabriel Taubin. Shield fields: Modeling and capturing 3d occluders. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 27(5), 2008. 8

[23] Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul Debevec. Deeplight: Learning illumination for unconstrained mobile mixed reality. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[24] Chloe LeGendre, Wan-Chun Ma, Rohit Pandey, Sean Fanello, Christoph Rhemann, Jason Dourgarian, Jay Busch, and Paul Debevec. Learning illumination from diverse portraits. In *SIGGRAPH Asia 2020 Technical Communications*, SA '20, New York, NY, USA, 2020. Association for Computing Machinery. 2

[25] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 2

[26] Yuanzhen Li, Hanqing Lu, Heung-Yeung Shum, et al. Multiple-cue illumination estimation in textured scenes. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1366–1373. IEEE, 2003. 2

[27] Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.*, 38(6), Nov. 2019. 2, 4

[28] 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

[29] John Murray-Bruce, Charles Saunders, and Vivek K. Goyal. Occlusion-based computational periscopy with consumer cameras. In Dimitri Van De Ville, Manos Papadakis, and Yue M. Lu, editors, *Wavelets and Sparsity XVIII*, volume 11138, pages 286 – 297. International Society for Optics and Photonics, SPIE, 2019. 2

[30] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), Dec. 2019. 2, 4

[31] Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. Seeing the world in a bag of chips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1417–1427, 2020. 2

[32] Gustavo Patow and Xavier Pueyo. A survey of inverse rendering problems. *Computer Graphics Forum*, 22(4):663–687, 2003. 2

[33] Joshua Rapp, Charles Saunders, Julián Tachella, et al. Seeing around corners with edge-resolved transient imaging. *Nature Communications*, 11(5929), Nov. 2020. 2

[34] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination from shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):290–300, 2003. 2

[35] Charles Saunders, John Murray-Bruce, and Vivek K Goyal. Computational periscopy with an ordinary digital camera. *Nature*, 565(7740):472–475, Jan. 2019. 2

[36] S. W. Seidel, Y. Ma, J. Murray-Bruce, C. Saunders, W. T. Freeman, C. C. Yu, and V. K. Goyal. Corner occluder computational periscopy: Estimating a hidden scene from a single photograph. In *2019 IEEE International Conference on Computational Photography (ICCP)*, pages 1–9, May 2019. 2, 4, 6, 7

[37] 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. 2

[38] Pratul P. Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T. Barron, Richard Tucker, and Noah Snavely. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *CVPR*, 2020. 2

[39] Matthew Tancik, Guy Satat, and Ramesh Raskar. Flash photography for data-driven hidden scene recovery. *arXiv preprint arXiv:1810.11710*, 2018. 2

[40] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. State of the art on neural rendering. *EG*, 2020. 2

[41] C. Thrampoulidis, G. Shulkind, F. Xu, W. T. Freeman, J. H. Shapiro, A. Torralba, F. N. C. Wong, and G. W. Wornell. Exploiting occlusion in non-line-of-sight active imaging. *IEEE Transactions on Computational Imaging*, 4(3):419–431, Sep. 2018. 2

[42] Antonio Torralba and William T. Freeman. Accidental pinhole and pinspeck cameras. *International Journal of Computer Vision*, 110(2):92–112, Mar. 2014. 2

[43] Henrique Weber, Donald Prévost, and Jean-François Lalonde. Learning to estimate indoor lighting from 3d objects. *2018 International Conference on 3D Vision (3DV)*, pages 199–207, 2018. 2

[44] Y. Yang and A. Yuille. Sources from shading. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 534–539, June 1991. 2

[45] A. B. Yedidia, M. Baradad, C. Thrampoulidis, W. T. Freeman, and G. W. Wornell. Using unknown occluders to recover hidden scenes. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12223–12231, June 2019. 2

[46] L. Zhang, Q. Yan, Z. Liu, H. Zou, and C. Xiao. Illumination decomposition for photograph with multiple light sources. *IEEE Transactions on Image Processing*, 26(9):4114–4127, 2017. 2